Dokumen Teknis

Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge* Menggunakan Arsitektur *Microservice*

Tugas Proyek Akhir Semester

Mata Kuliah: Pengembangan Aplikasi Terdistribusi

Dipersiapkan oleh:

NIM 11322005	Maria Elimadona Sibarani
NIM 11322015	Rivael Hasiholan Manurung
NIM 11322031	Daniel Pandapotan Manalu
NIM 11322044	Kristina Sitorus

Untuk:

Institut Teknologi Del

2024

DAFTAR ISI

1 Pendahuluan 1
1.1 Deskripsi Umum Aplikasi 1
1.2 Karakteristik Pengguna Aplikasi 5
1.3 Fungsi pada Aplikasi 6
2 Desain Rancangan Aplikasi 9
2.1 Autentikasi 9
2.2 Product 11
2.3 Category 13
2.4 Table 15
2.5 Order 18

1 Pendahuluan

Pada bab 1 berisi tentang Deskripsi Umum aplikasi, Karakteristik Pengguna Aplikasi, dan Fungsi pada Aplikasi.

1.1 Deskripsi Umum Aplikasi

Pembangunan aplikasi berbasis *mobile* ini ditujukan untuk memanajemen dan melakukan reservasi secara *online*. Aplikasi ini dapat digunakan oleh *admin* dan *customer* untuk memasarkan dan membeli produk. Pada aplikasi ini, *admin* akan mengelola produk yang akan dijual, sedangkan *customer* dapat melakukan pemesanan terhadap produk tersebut.

Aplikasi ini dibangun dengan menggunakan 3 bahasa pemrograman. Bagian *back-end* dan *front-end* dibangun dengan bahasa yang berbeda. Untuk bagian *back-end* digunakan bahasa *GO* dan untuk bagian *front-end* menggunakan bahasa *PHP* (*Admin), DART* (*Customer)*. Selain bahasa pemograman aplikasi ini juga dibangun dengan teknologi *Flutter, GOFiber, Apache,* dan *Laravel*. Teknologi disamping akan dijelaskan lebih *detail* dibawah ini:

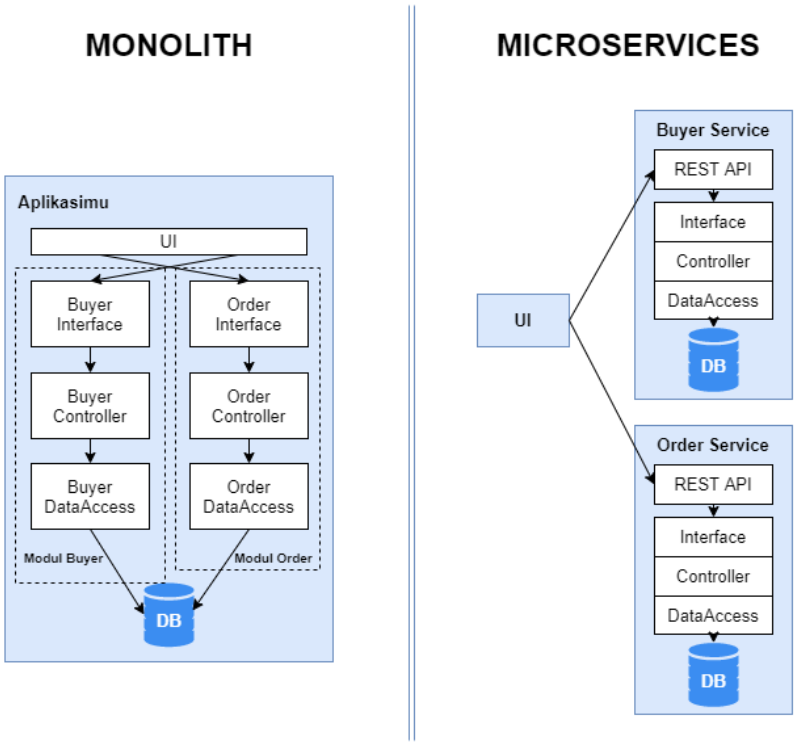
1. *Flutter* yang berfungsi sebagai *framework* yang digunakan dalam pembangunan aplikasi dengan bahasa pemograman *DART*.
2. *GoFiber* yang berfungsi sebagai *framework* yang digunakan untuk membangun aplikasi dengan bahasa pemograman *GO* atau Golang.
3. *Apache* yang berfungsi sebagai *web server* yang digunakan untuk melayani permintaan HTTP dan mengirimkan konten *web* kepada pengguna berdasarkan permintaan.
4. *Laravel* yang berfungsi sebagai *framework* yang digunakan dalam pembangunan aplikasi dengan bahasa pemograman *PHP*.

Aplikasi ini juga dibangun dengan menggunakan beberapa *tools* diantaranya:

1. Teks Editor/IDE : *Visual Studio Code*
2. *Frontend Development* : *Flutter, Laravel*
3. *Backend Development* : *GOFiber*
4. *Version Control System* : *GitHub*
5. *Database Management System* : *MySQL*

6. *API Testing* : *Postman*
7. *Development Environment* : *XAMPP*

Aplikasi ini menggunakan arsitektur *microservice*. Sesuai dengan namanya, arsitektur ini dirancang pada sebuah sistem untuk membagi *service* menjadi *service* yang lebih kecil. Lalu mengapa tidak menggunakan arsitektur *monolith?*, Kita akan membahas perbedaan *monolith dan microservices.*

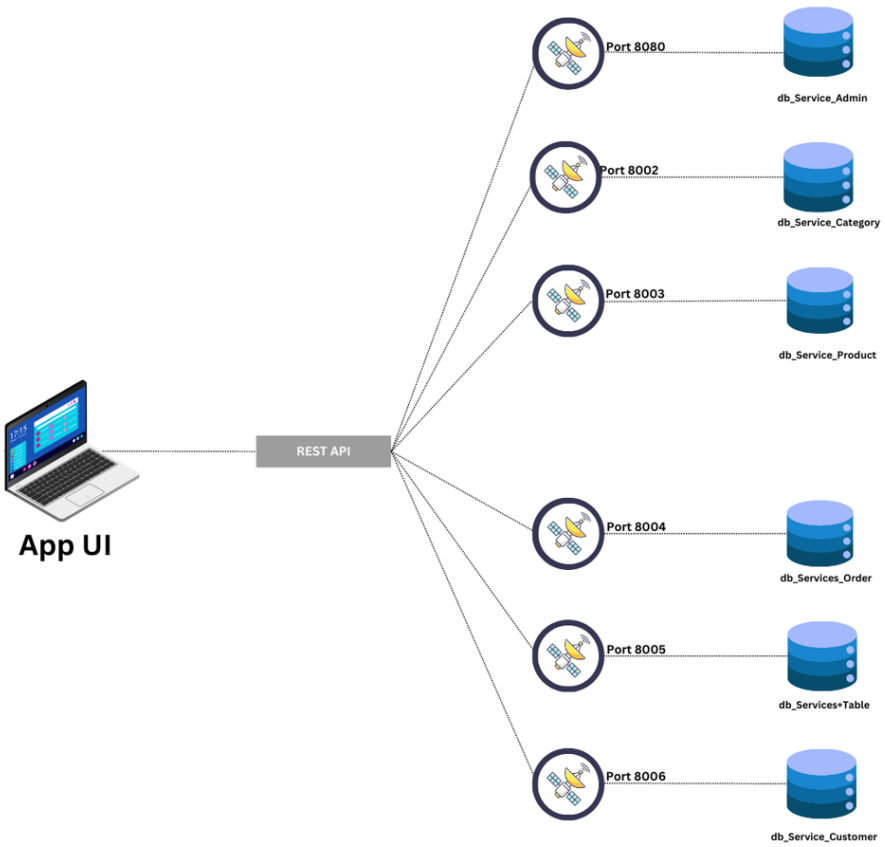


Gambar 1 Perbandingan Monolith dengan Microservices
Sumber : https://kutu.dev/artikel/microservice

Untuk memahami perbedaan arsitektur *monolith* dengan *microservices*, kita harus terlebih dahulu memahami pengertian dari masing-masing arsitektur. *Microservice* merupakan arsitektur yang condong kepada banyak *serviceI atau aplikasi yang tidak terkait erat satu dengan yang lainnya (Loosely coupled)*. *Monolith* merupakan arsitektur yang dimana keseluruhan kode akan dikomplikasi menjadi satu aplikasi yang dimana aplikasi tersebut menjalankan seluruh proses yang dibutuhkan. Pada *monolith* kebutuhan akan komunikasi kepada aplikasi atau *service* lain bisa jadi tidak ada, karena aplikasi telah mencakup seluruh kode yang dibutuhkan. Untuk lebih

lanjut mari beralih pada cakupan setiap arsitektur. Pada *monolith* fungsionalitas yang dibutuhkan diimplementasikan secara lengkap dan menyeluruh pada satu *codebase,* sedangkan pada *microservice* satu *service* focus hanya pada satu fungsi. Sekarang lihat perbedaan melalui jumlah. Pada *monolith* jumlah biasanya hanya menjadi satu artefak atau binary atau *service*, sedangkan pada microservice dibutuhkan banyak *service* atau aplikasi yang saling berkomunikasi. Selanjutnya perbedaan dari segi komunikasi antar modul. Pada *monolith* komunikasi antar modul dilakukan melalui *function call* pada kode tanpa perlu protocol eksternal khusus, sedangkan pada *microservice* diperlukan protocol komunikasi yang ringan agar komunikasi antar *service* bisa berjalan baik. Yang terakhir adalah dari sisi *deployment.* Pada *monolith deployment* biasanya lebih lambat karena perlu men-*deploy* kode yang besar pada satu waktu yang tidak fleksibel, sedangkan pada *microservice deployment* lebih fleksibel, cepat, dan dapat dilakukan secara terpisah untuk masing masing *service*.

Dengan ini, arsitektur *microservice* lebih memungkinkan pengembangan yang lebih fleksibel, dan efisien dibandingkan *monolith* maka setiap *service* pada proyek ini akan memiliki *database*-nya masing-masing. *Service* yang ada pada aplikasi adalah *Admin Service, Category Service,* dan *Customer Service, Order Service, Product Service, Table Service*. Setiap *service* akan berjalan secara *independent* sehingga satu *service* tidak akan memengaruhi *service* lainnya. Namun, meskipun berdiri sendiri, setiap *service* ini tetap saling terhubung antara satu dengan yang lainnya. Antar-*service* ini akan saling berkomunikasi dengan HTTP *Request* berstandar REST API. Setiap *database* pada masing-masing *service* memiliki *port* yang sama. Namun, *port* yang digunakan pada setiap *service* tersebut akan berbeda. Hal ini ditujukan sebagai pemisah antar-*service* sehingga saat satu *service* sedang tidak dapat dijalankan, maka *service* yang lainnya masih tetap dapat berjalan.



Gambar 2 Arsitektur *Microservices*

Pada Gambar 1 terlampir arsitektur *microservice* yang digunakan pada Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge*. REST API digunakan sebagai penghubung antara *APP UI* dengan *service* yang tersedia sehingga memungkinkan *service* dapat dikonsumsi oleh *APP UI*. Hal inilah yang menyebabkan *service* dapat diakses oleh beberapa *platform* tanpa harus terbatas pada satu bahasa pemrograman saja. Sama halnya dengan Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge* ini. Bahasa yang digunakan pada bagian *front-end* adalah bahasa *PHP* dan *DART* dan pada bagian *back- end* menggunakan bahasa *GO*. Meskipun memiliki bahasa yang berbeda, tetapi aplikasi akan tetap dapat berjalan dengan adanya REST API. Pada bagian *back-end*, setiap *service* memiliki *database*-nya masing-masing. Pada Aplikasi ini terdapat enam *service* dan enam *database. Service* yang tersedia adalah *admin service* dengan *database service_admin, category service* dengan *database service_category, customer service* dengan *database service_customer, order service* dengan *database service_order, product service* dengan *database service_product* dan *table service* dengan *database service_table*.

Setiap *service* dirancang memiliki *port* yang berbeda sehingga *port* tidak akan bertabrakan.

1.2 Karakteristik Pengguna Aplikasi

Pada Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge* ini, terdapat dua kategori pengguna. Kategori pengguna pada aplikasi adalah *admin* dan *customer*. Karakteristik pengguna aplikasi dilampirkan pada Tabel 1.

Table 1 Karakteristik Pengguna Aplikasi

Kategori Pengguna	Fungsi	Hak Akses ke Aplikasi
Admin	Mengelola data dalam *website*	1. Akses ke menu *login* 2. Akses ke fungsi penambahan, pengubahan, dan penghapusan data produk 3. Akses ke fungsi penambahan, pengubahan, dan penghapusan data table 4. Akses ke fungsi penambahan, pengubahan, dan penghapusan data kategori 5. Akses untuk keluar sistem melalui *logout*
Customer	Melakukan *order* melalui *mobile*	1. Akses ke menu registrasi 2. Akses ke menu *login* 3. Akses ke fungsi melihat produk 4. Akses ke fungsi melihat kategori 5. Akses ke fungsi melihat data 6. Akses ke fungsi perubahan data diri 7. Akses ke pemesanan produk 8. Akses untuk keluar sistem melalui *logout*

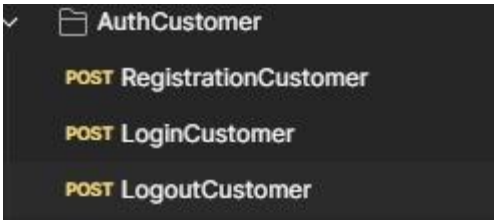
1.3 Fungsi pada Aplikasi

Adapun fungsi yang terdapat pada Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge*, yaitu :

1. **Fungsi Autentikasi**

   Fungsi autentikasi dapat dilakukan oleh *Admin* dan *Customer*. Fungsi autentikasi terdiri dari 1 *method* yaitu *post. Method post* digunakan untuk melakukan *login* (*Login, LoginCustomer),* melakukan *logout* (*Logout_Admin, LogoutCustomer)* dan melakukan *registration (RegistrationCustomer).*

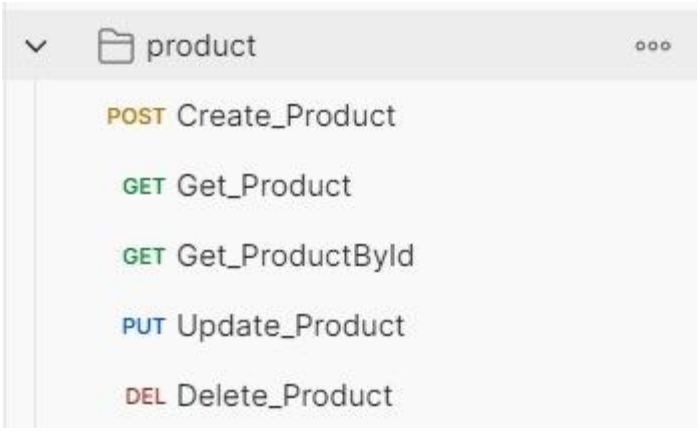
Gambar 3 Method Autentikasi Admin



Gambar 4 Method Autentikasi Customer

2. **Fungsi Mengelola *Product Admin***

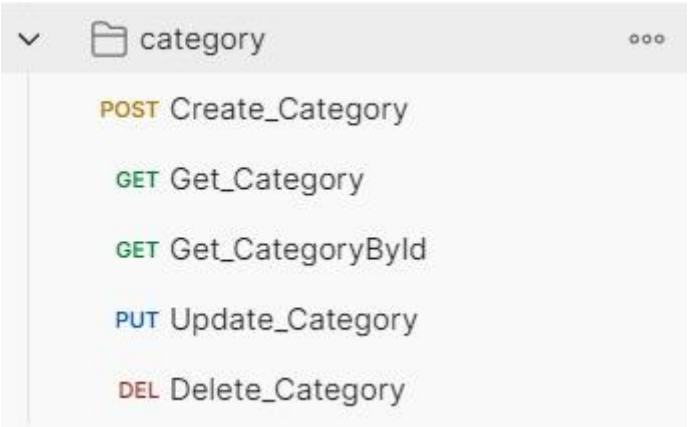
   Fungsi mengelola produk digunakan oleh *Admin* untuk menambah, menghapus, mengedit, dan melihat produk yang dibagi berdasarkan kategori yang telah dibuat sebelumnya pada Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge*. Fungsi mengelola *product* terdiri dari 4 *method* yaitu *post, get, put* dan *delete*. *Method post* digunakan untuk menambahkan *product* (*Create_Product), method get* digunakan untuk mengambil semua data *product (Get_Product)* dan mengambil data *product* berdasarkan *id (Get_ProductByid), method put* digunakan untuk mengedit data *product (Update_Product), method delete* digunakan untuk menghapus produk (*Delete_Product).*



Gambar 5 Method Mengelola Product Admin

3. Fungsi Mengelola *Category Admin*

Fungsi Mengelola *Category* digunakan oleh *Admin* untuk menambah, menghapus, mengedit, dan melihat produk yang dibagi berdasarkan *category* yang telah dibuat sebelumnya pada Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge*. Fungsi mengelola *category* terdiri dari 4 *method* yaitu *post, get, put,* dan *delete*. *Method post* digunakan untuk menambahkan *category product (Create_Category), method get* digunakan untuk mengambil semua data *category product* (*Get_Category)* dan mengambil data *category product* berdasarkan *id (Get_CategoryById), method put* digunakan untuk mengedit data *category product (Update_Category), method delete* digunakan untuk menghapus data *category product (Delete_Category).*



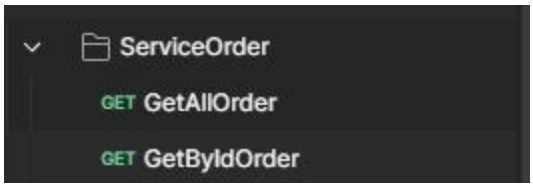
Gambar 6 Method Mengelola Category Admin

4. Fungsi Mengelola *Table Admin*

Fungsi Mengelola *Table* digunakan oleh *Admin* untuk menambah, menghapus, mengedit, dan melihat *table*. Fungsi mengelola *table* terdiri dari 4 *method.*

5. Fungsi Melihat *Order Admin*

Fungsi Melihat *Order* digunakan oleh *Admin* untuk melihat *order* dengan method *get*. *Method get* digunakan untuk mengambil data *order product* secara keseluruhan maupun berdasarkan *id* agar dapat dilihat oleh *admin.*



Gambar 7 Method Melihat Order Admin

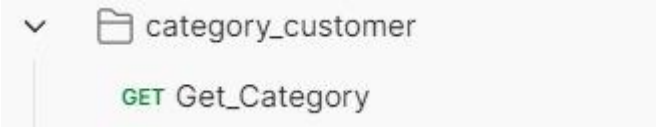
6. Fungsi Melihat *Product Customer*

Fungsi Melihat *Product* digunakan oleh *Customer* untuk melihat *product* dengan method *get*. *Method get* digunakan untuk mengambil data *product* agar dapat dilihat oleh *customer.*

Gambar 8 Method Melihat Product Customer

7. Fungsi Melihat *Category Customer*

Fungsi Melihat *Category* digunakan oleh *Customer* untuk melihat *category product* dengan method *get*. *Method get* digunakan untuk mengambil data *category* agar dapat dilihat oleh *customer*.



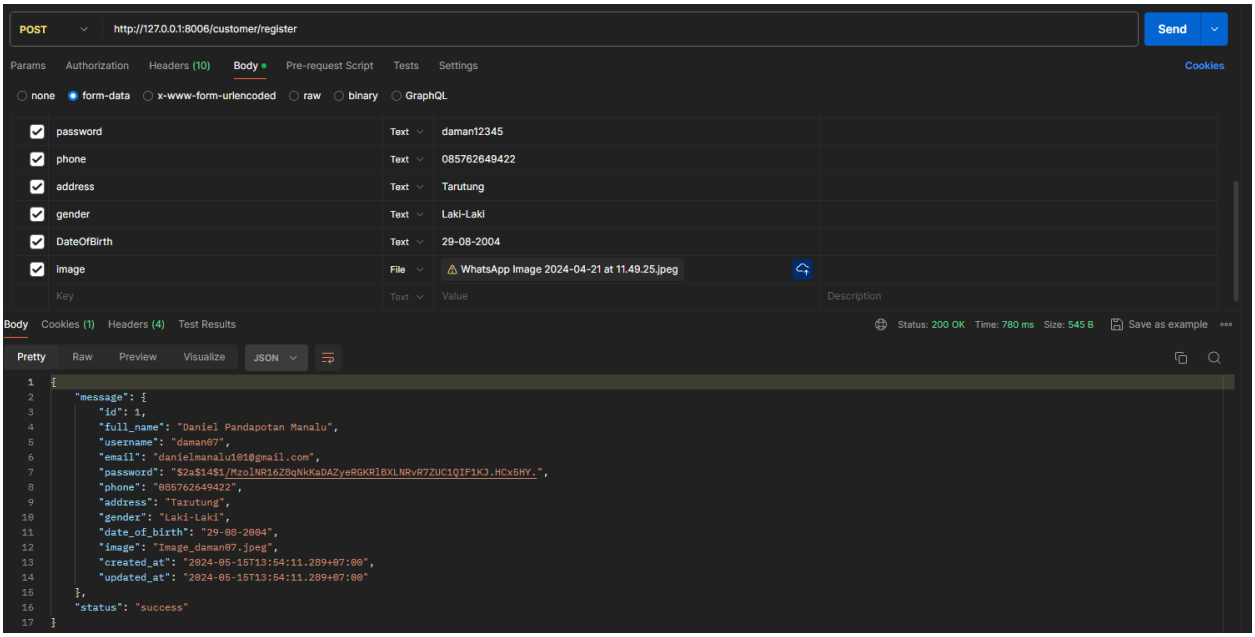
Gambar 9 Method Melihat Category Customer

2 Desain Rancangan Aplikasi

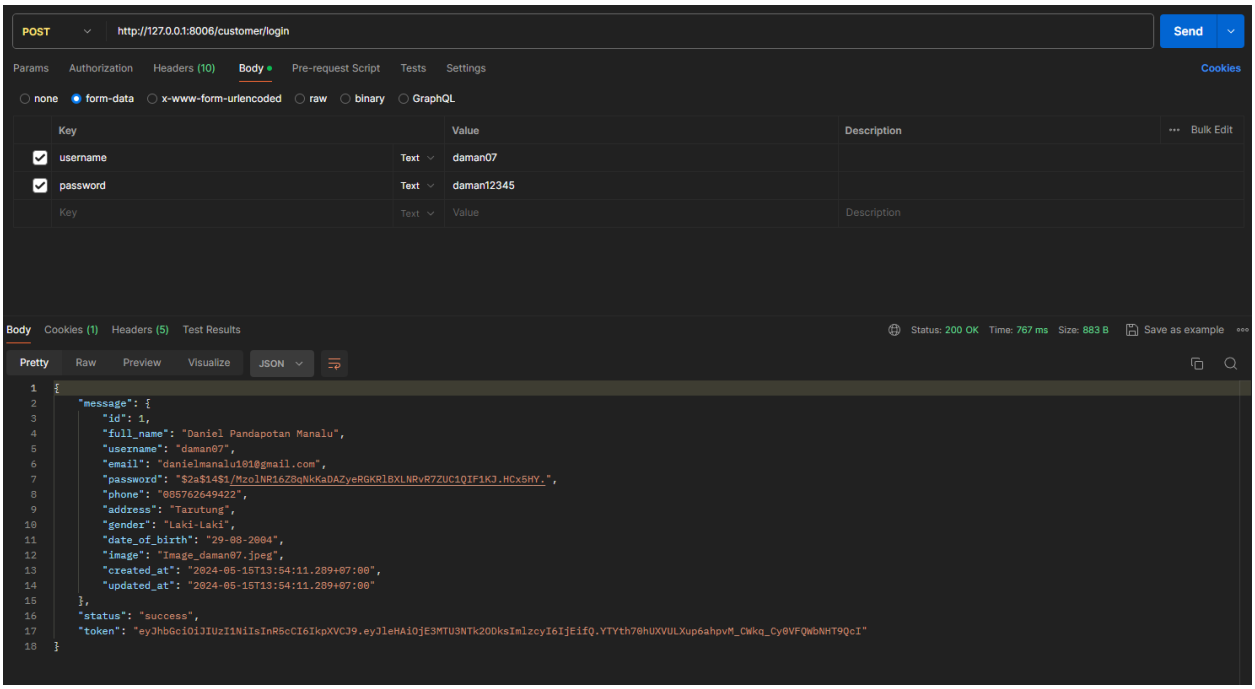
2.1 Autentikasi

a. Post

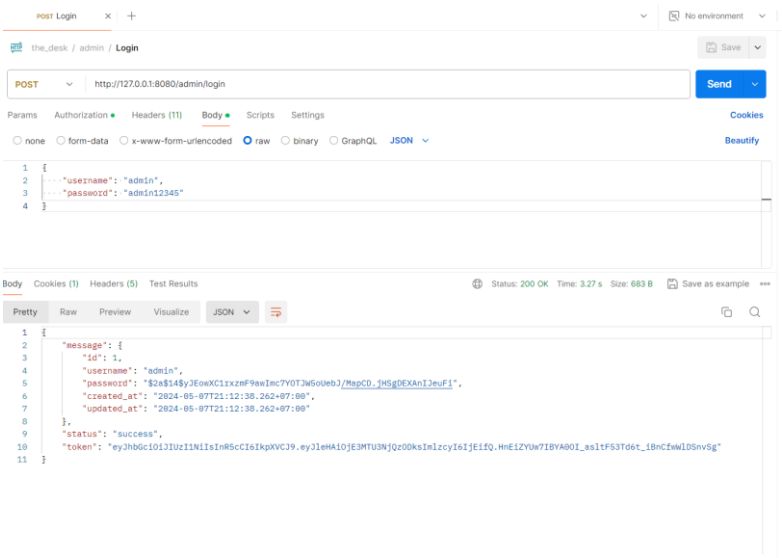
- Registrasi (*Customer)*



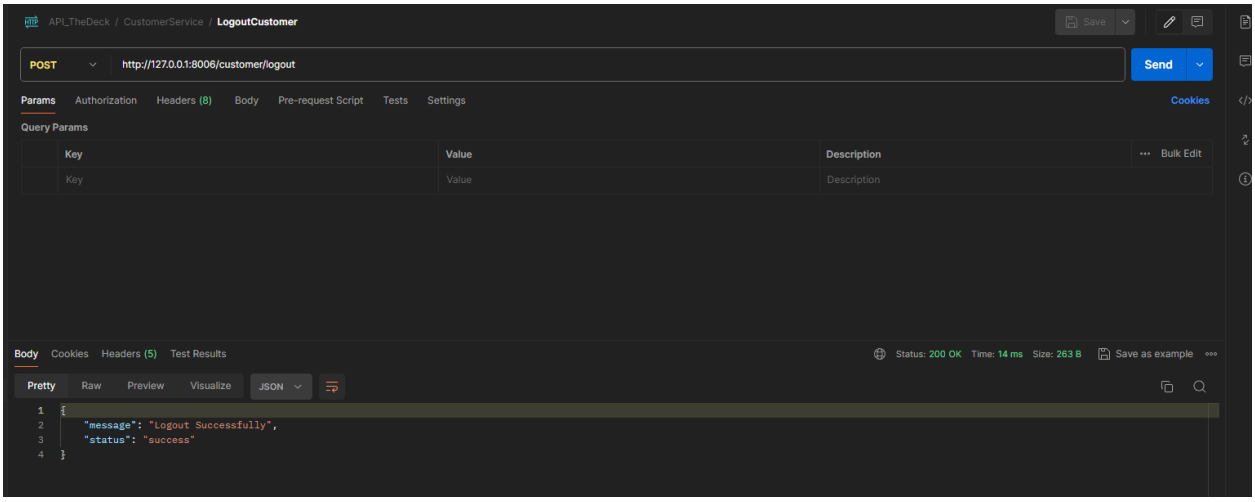
- Login (*Customer)*



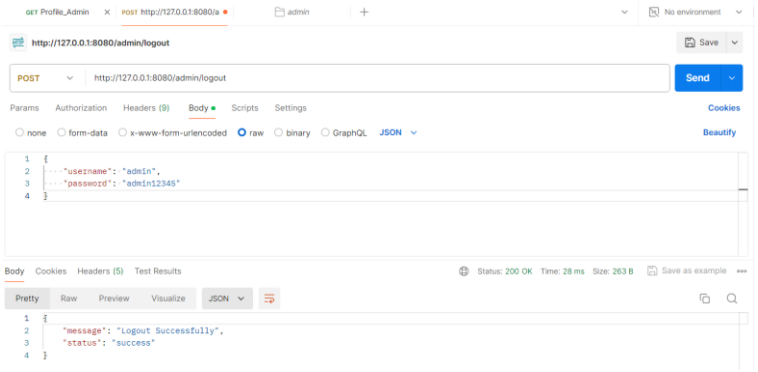
- Login (*Admin)*



- Logout (*Customer)*

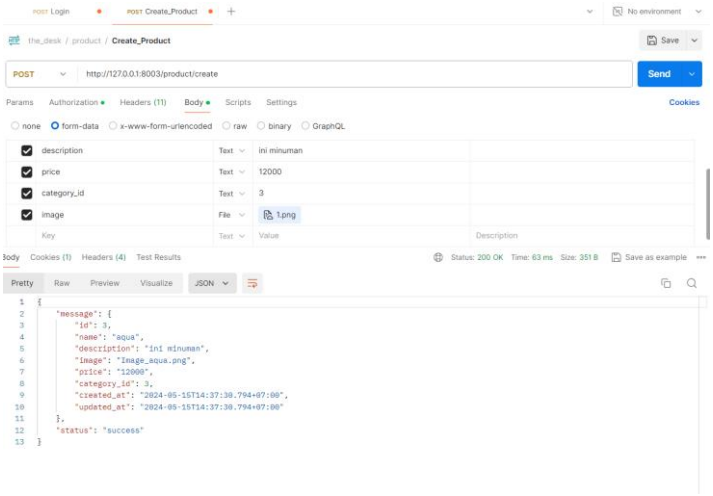


- Logout (*Admin)*



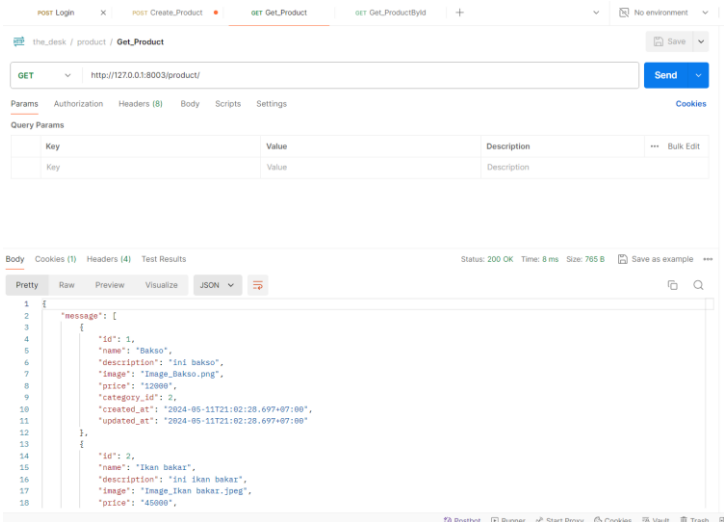
2.2 Product

a. Post

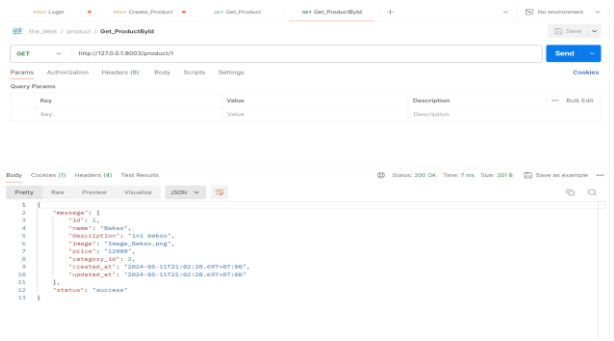


b. Get

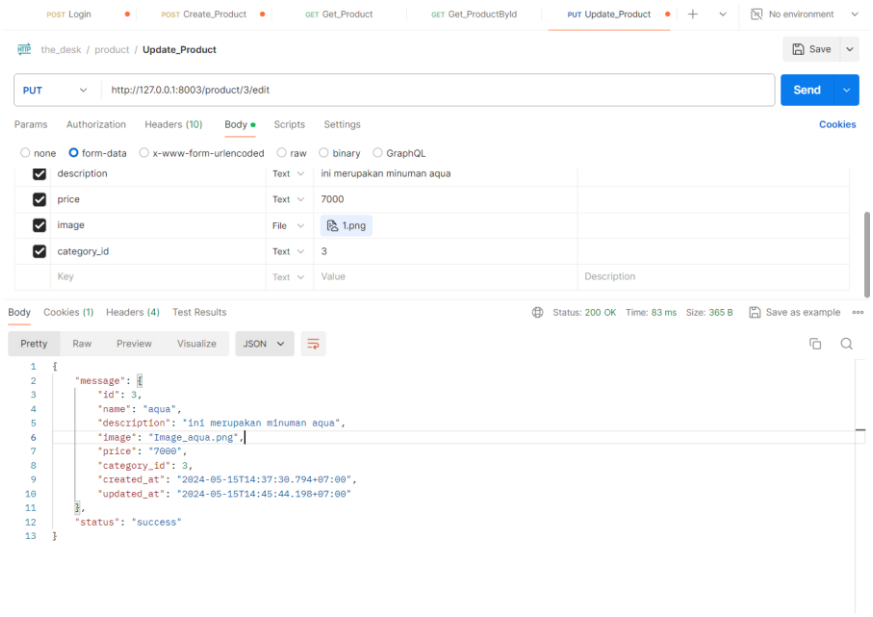
- Get Public



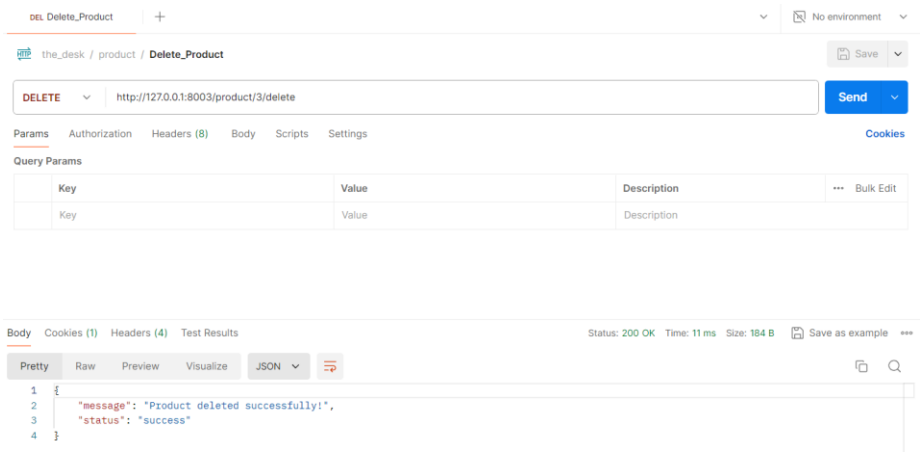
- Get Product by id



c. Put

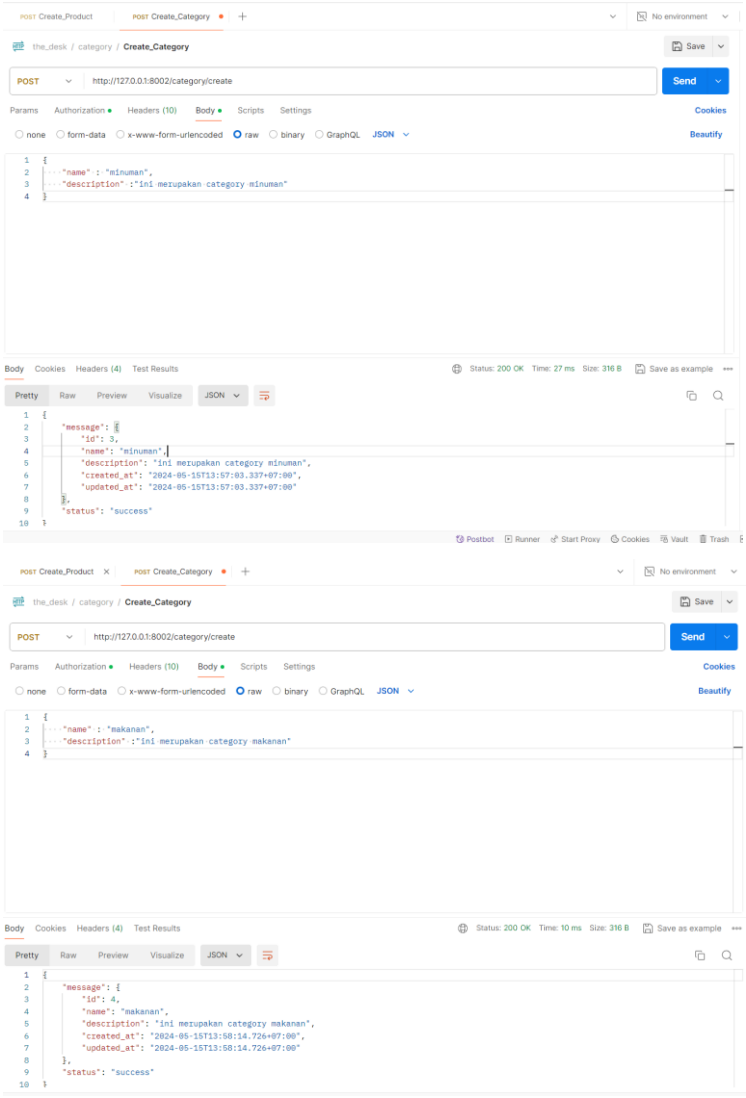


d. Delete



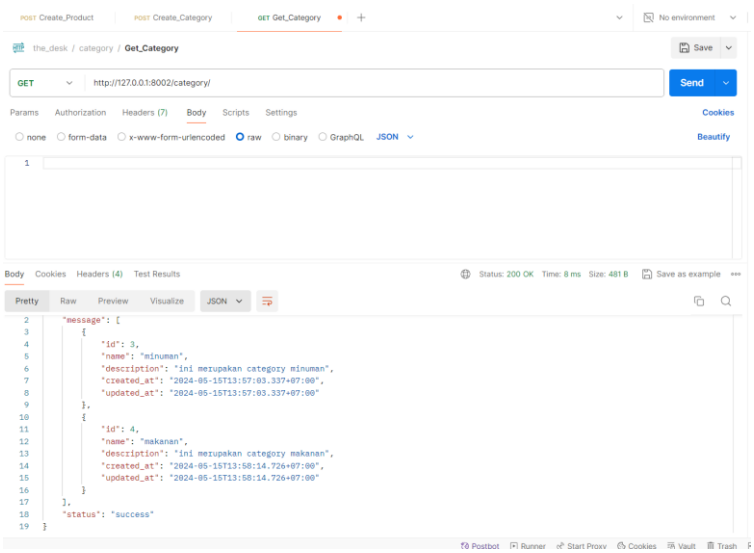
2.3 Category

a. Post

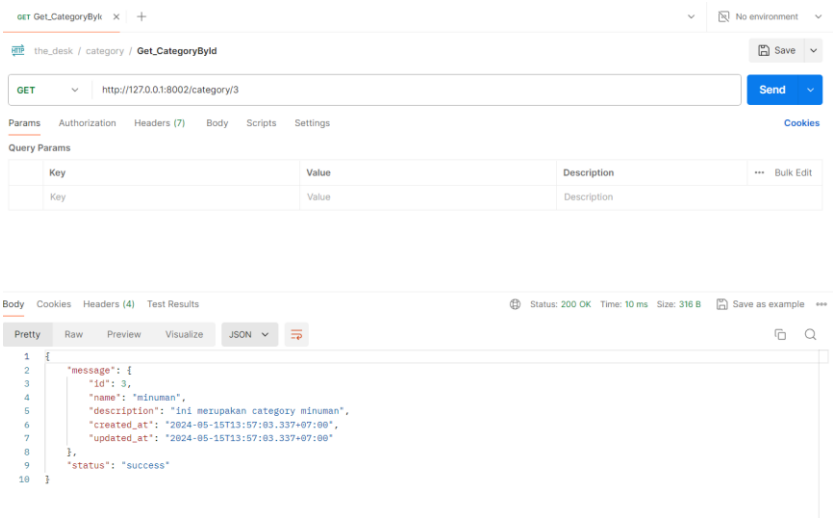


b. Get

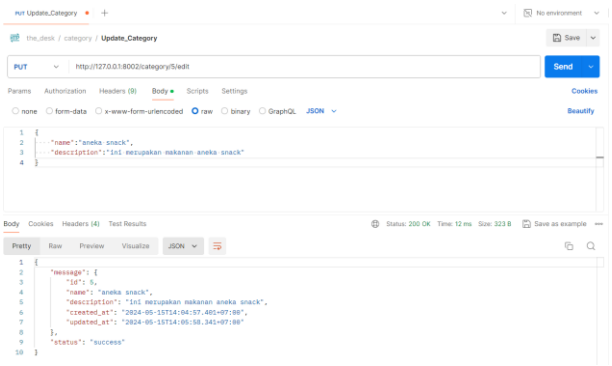
- Get Public



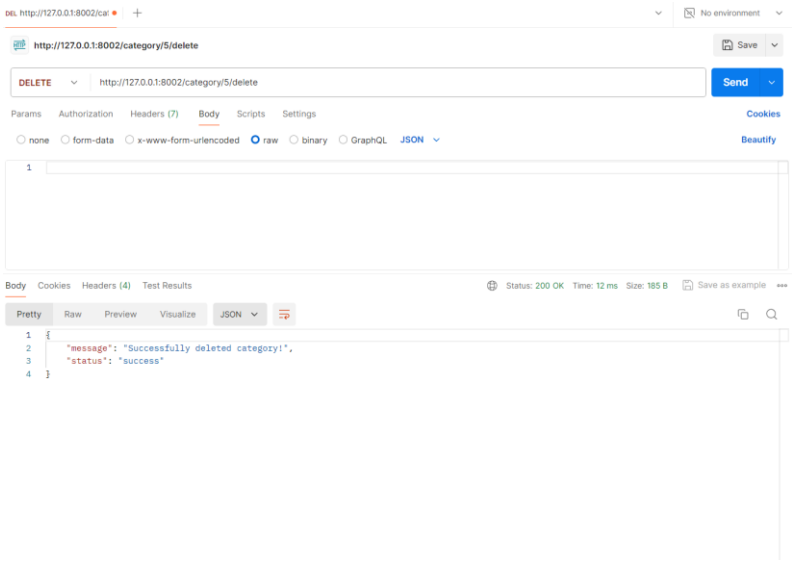
- Get Category by id



c. Put

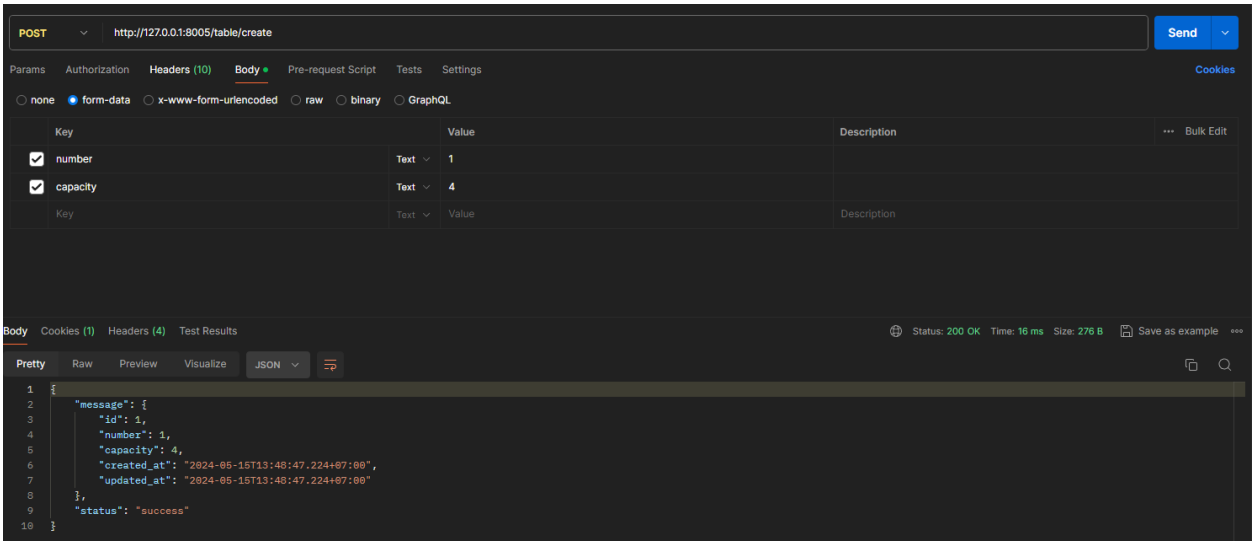


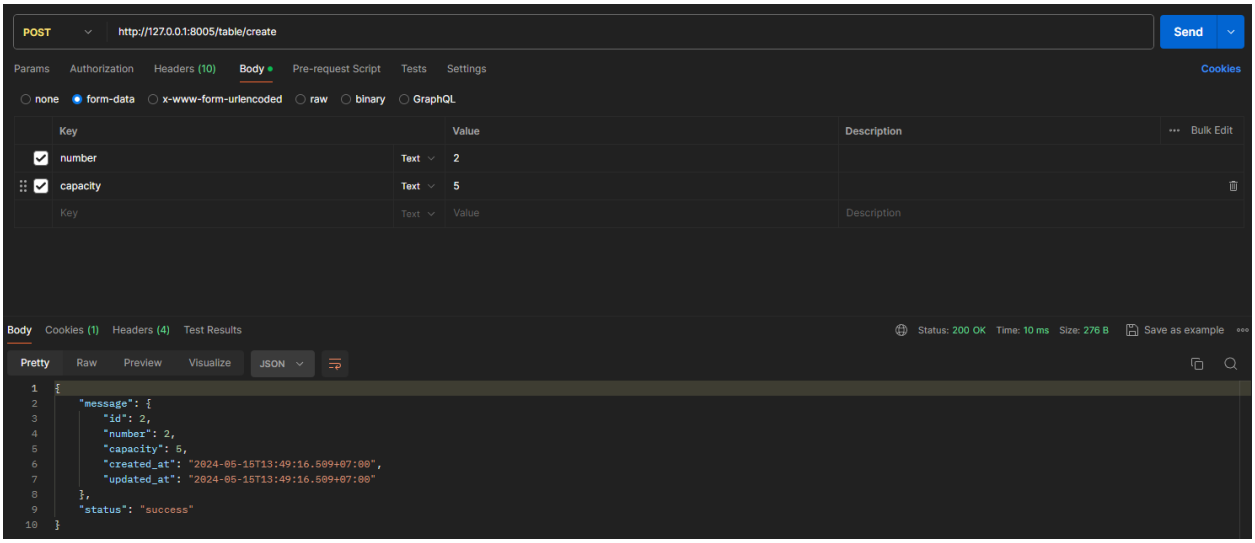
d. Delete



2.4 Table

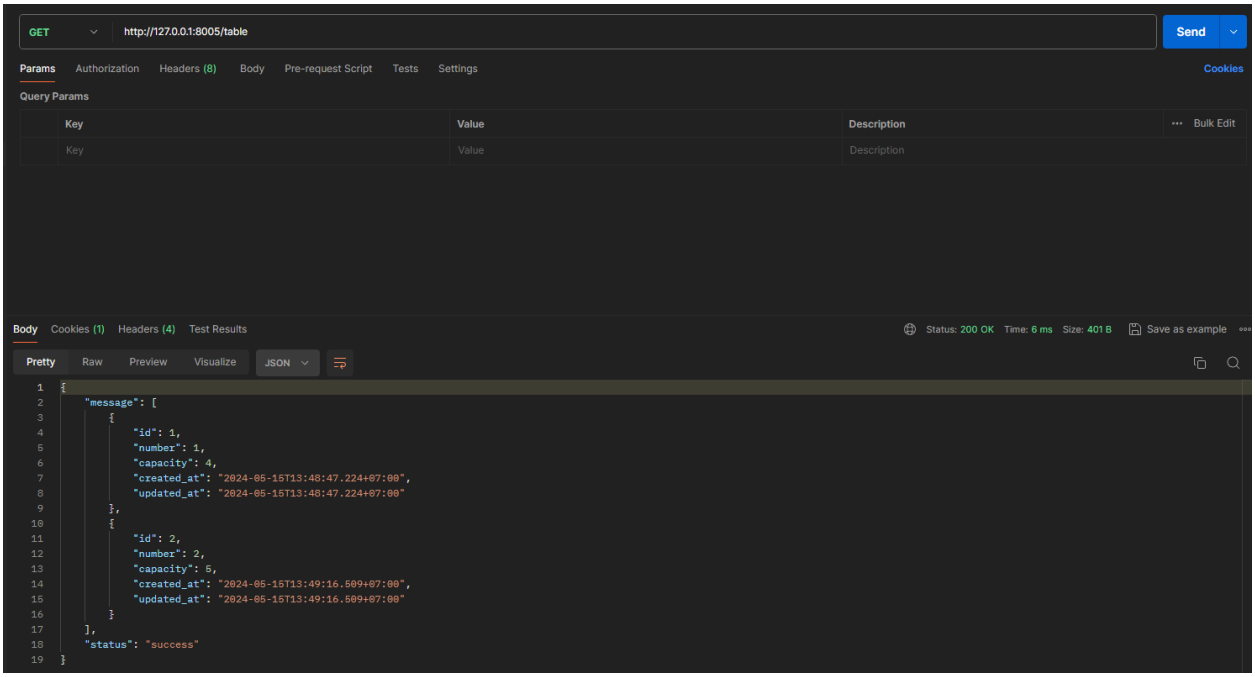
a. Post



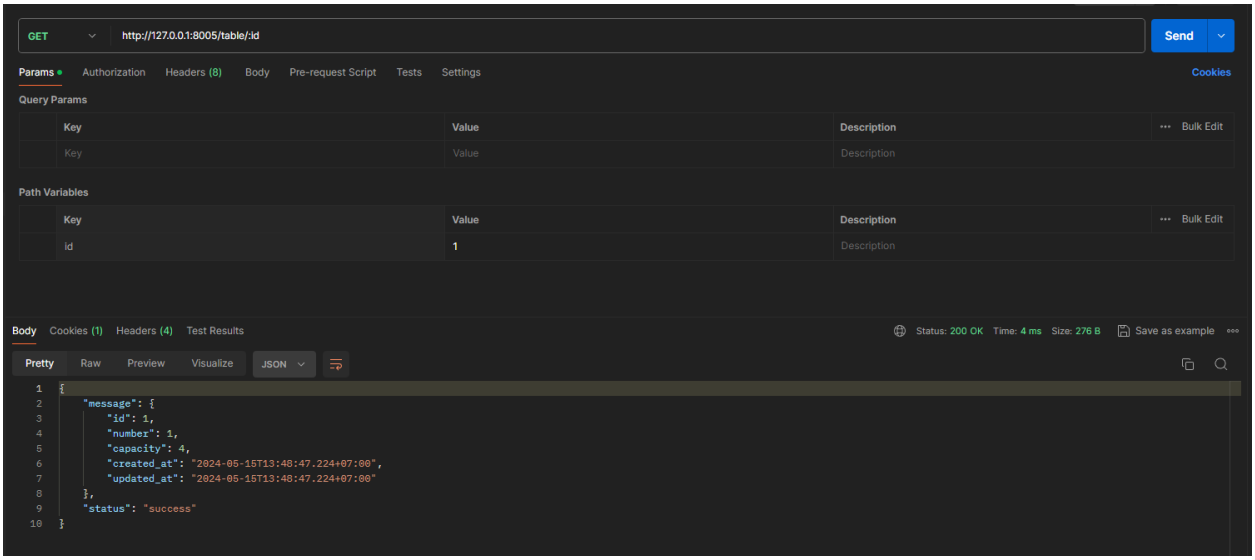


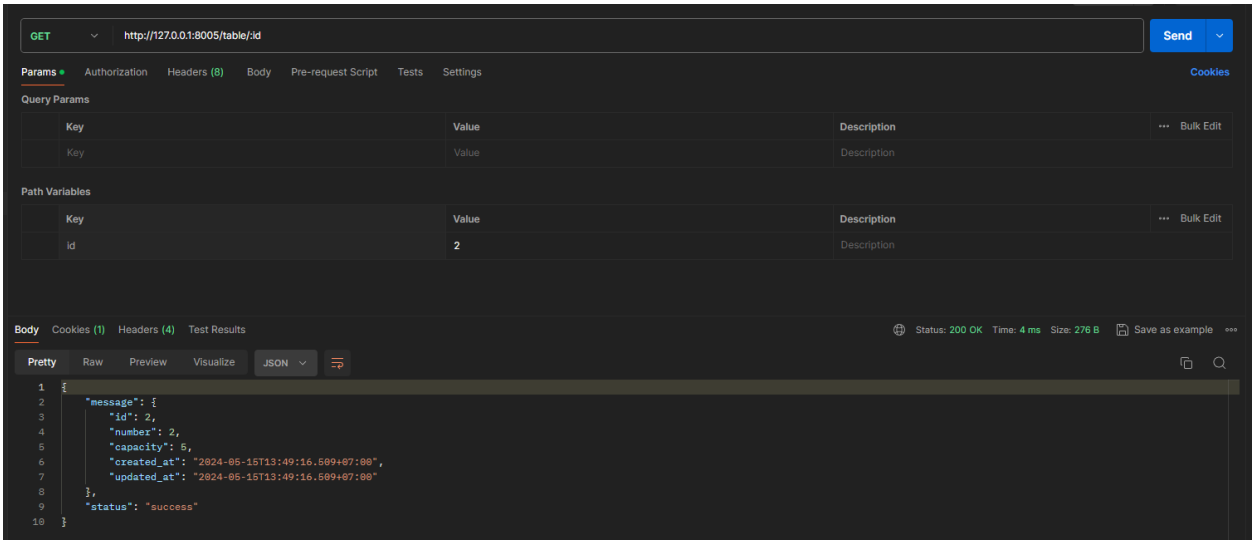
b. Get

- Get Public

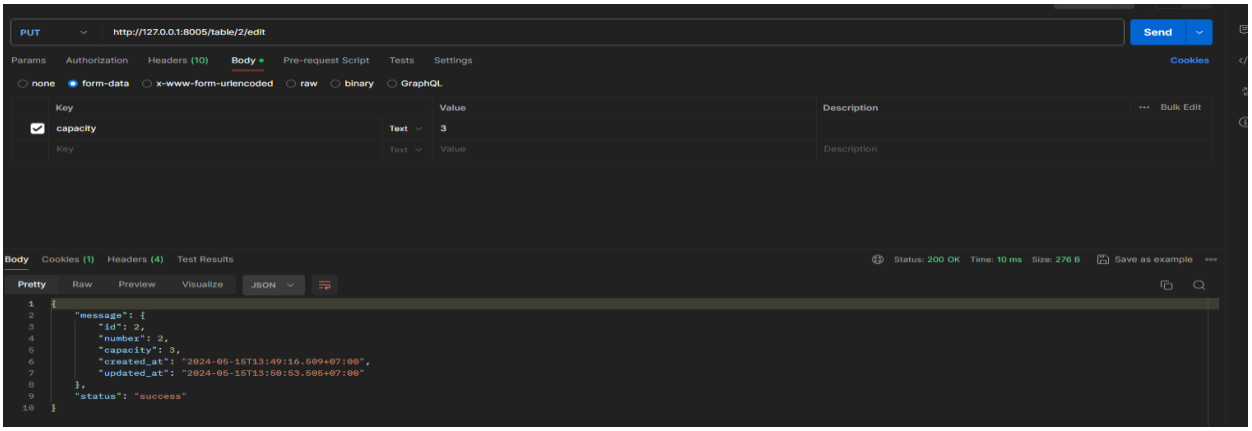


- Get Tables by id

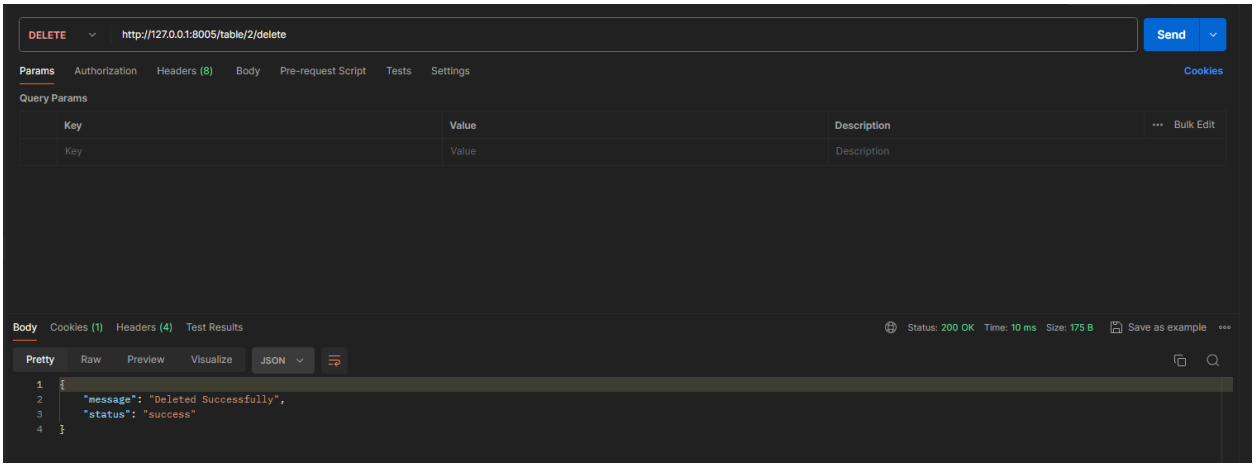




c. Put

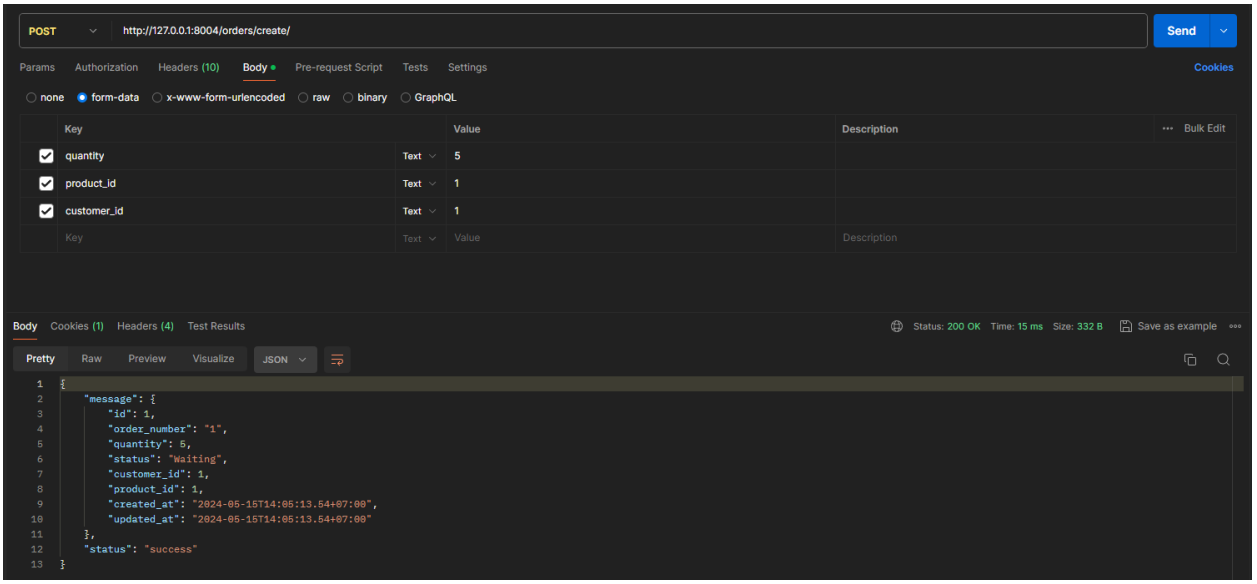


d. Delete



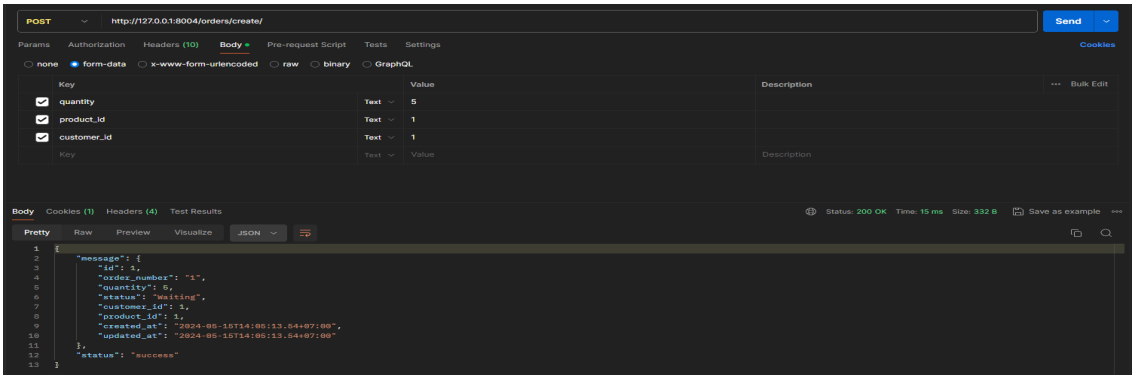
2.5 Order

a. Post

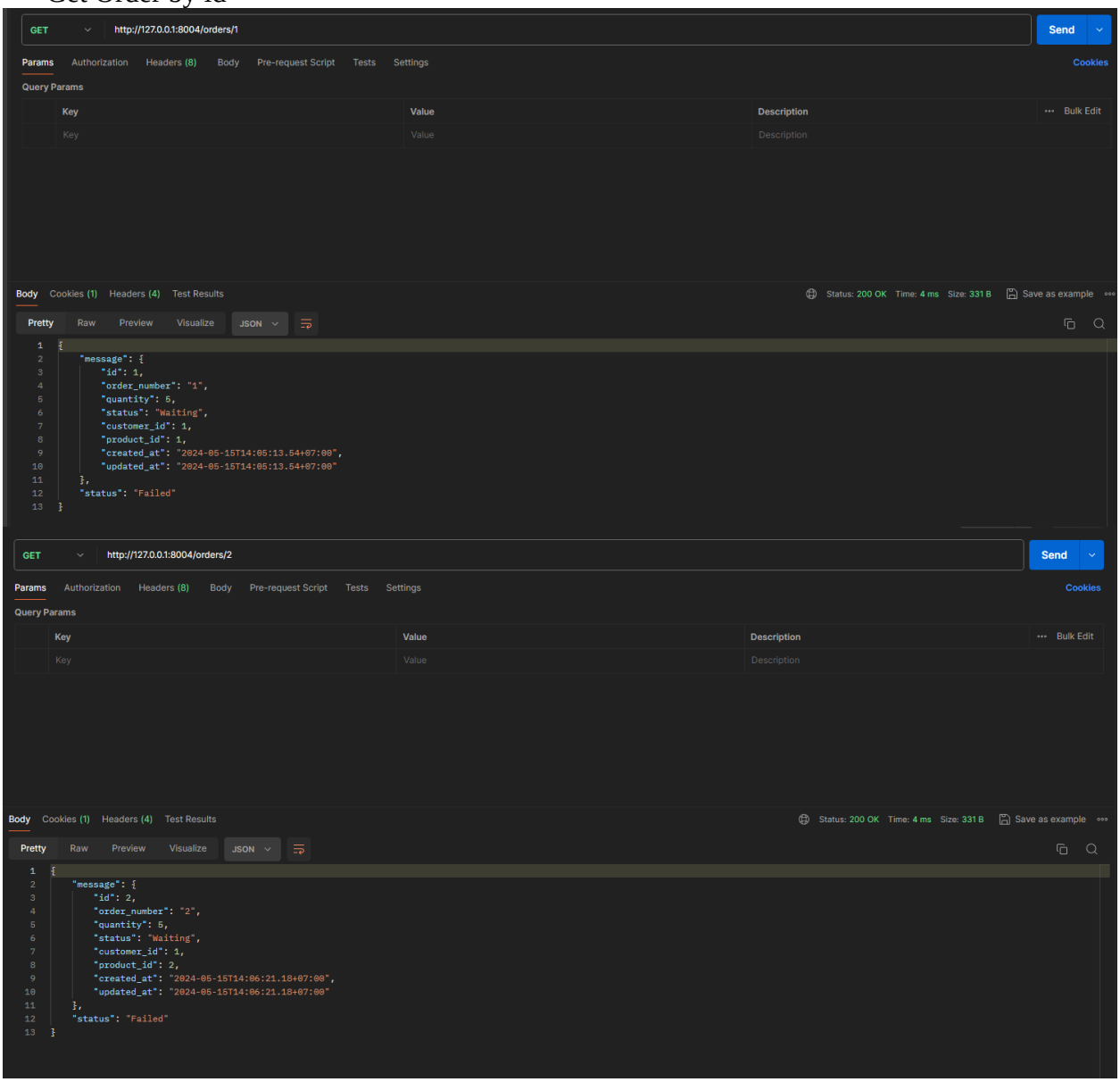


b. Get

- Get public



- Get Order by id



KESIMPULAN: Aplikasi Manajemen dan Reservasi *The Deck Restaurant and Lounge* adalah aplikasi berbasis *mobile* yang memungkinkan pengguna untuk melakukan manajemen dan reservasi secara online. Dibangun dengan menggunakan *Flutter, GOFiber, Apache,* dan *Laravel,* serta menggunakan arsitektur *microservice* untuk fleksibilitas dan efisiensi. Terdapat dua kategori pengguna: *admin* dan *customer*, masing-masing dengan hak akses yang berbeda. Fungsi aplikasi mencakup autentikasi, pengelolaan produk, kategori, meja, dan pesanan, dengan berbagai *method* yang tersedia. Desain rancangan aplikasi meliputi autentikasi, produk, kategori, meja, dan pesanan. Dengan demikian, aplikasi ini memungkinkan manajemen yang efisien dan pengalaman pemesanan yang lebih baik bagi pengguna.