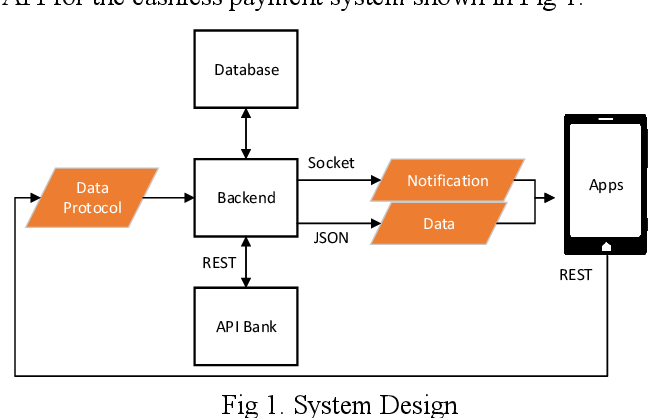
**Backend Engineer**

Nama : Muhammad Syaiful Maulana

1. Disebutkan seorang client membutuhkan layanan aplikasi antar makanan. Dan Anda diminta untuk ***membuat desain sistem*** dengan kriteria pelanggan hanya dilayani dengan *mobile apps*.
   1. Desainlah *stack backend dan micro service* layanan tersebut beserta *tools* dan alasan mengapa memilih *design* dan *tools* tersebut.

Jawab :



* Stack Progamming menggunakan MERN karena MERN menyediakan antarmuka pengguna berkinerja tinggi. Hemat biaya dan open source. Dapat dengan mudah beralih antara klien dan server.

1. Bagaimana menangani keamanan dalam pengiriman data (*backend* dan *mobile apps*) pada sistem diatas.
   1. Jelaskan solusi tersebut beserta alasannya.

Jawab :

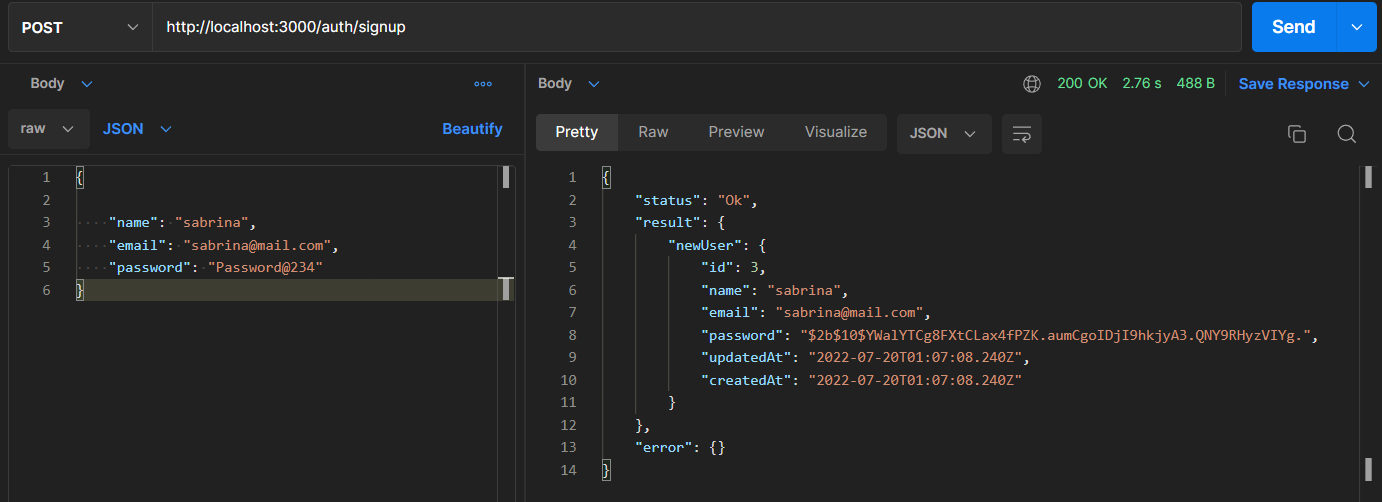
Untuk menangani mungkin bisa dilakukan beberapa cara, yaitu:

* Menggunakan database internal dan eksternal
* Melakukan double checking
* Membatasi akses

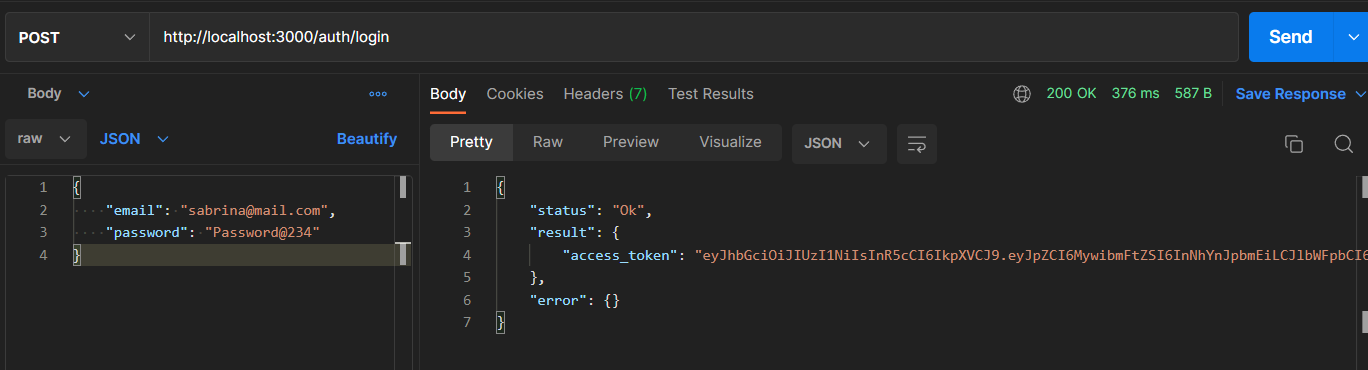
1. Buatlah sebuah *service RESTFul API* berdasarkan dokumentasi apiary berikut. Soal nomor 3 tidak ada hubungannya dengan soal nomor 1. <https://testbinar.docs.apiary.io/>. Boleh menggunakan bahasa apapun.
   1. Sertakan panduan cara instalasi tools, pengaturan database, penggunaan kode dan cara ujicoba kode dalam file Readme.md
   2. Sertakan tangkapan layar proses ujicoba API menggunakan postman.
   3. Kode yang baik adalah kode yang bisa dibaca dan dipakai oleh orang lain.

Jawaban :

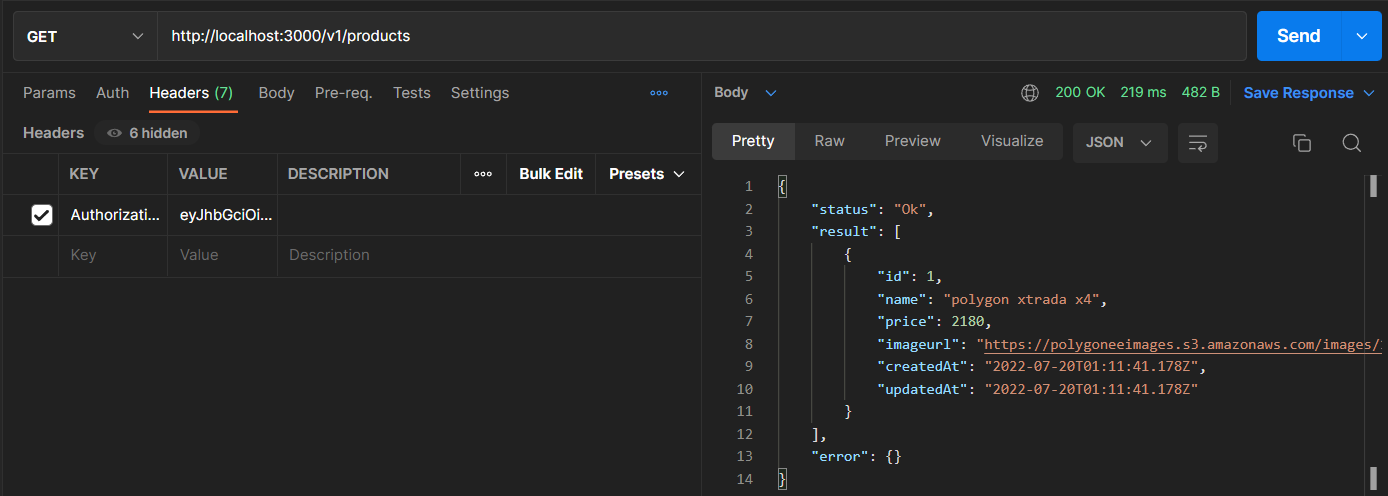
* Auth/signup



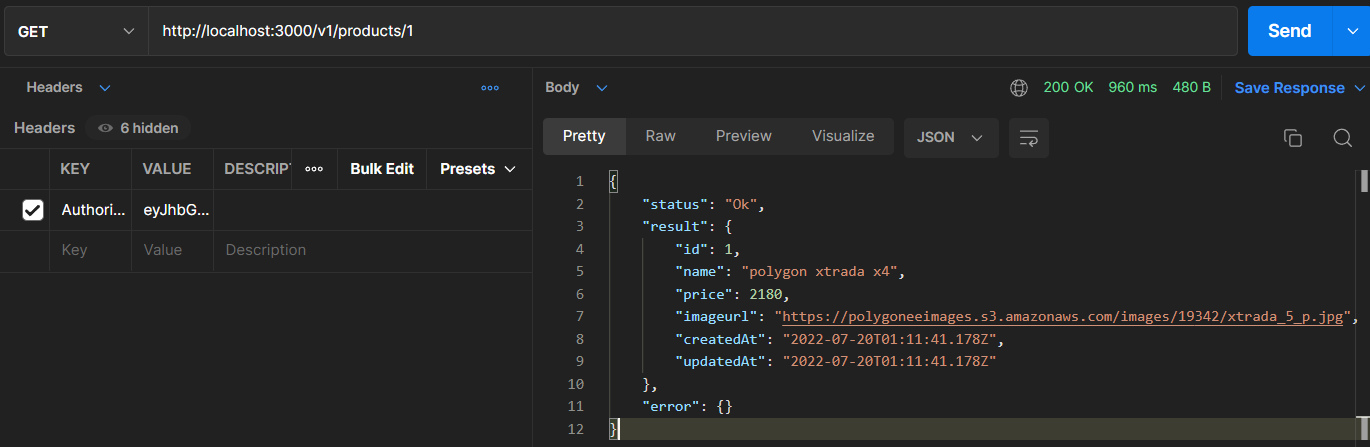
* Auth/login



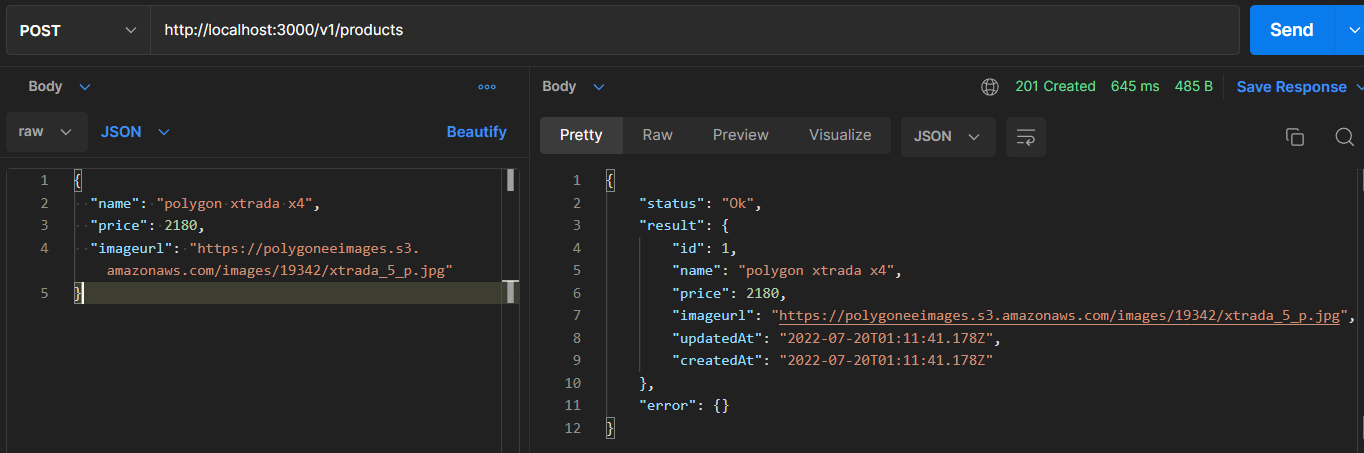
* Show Data



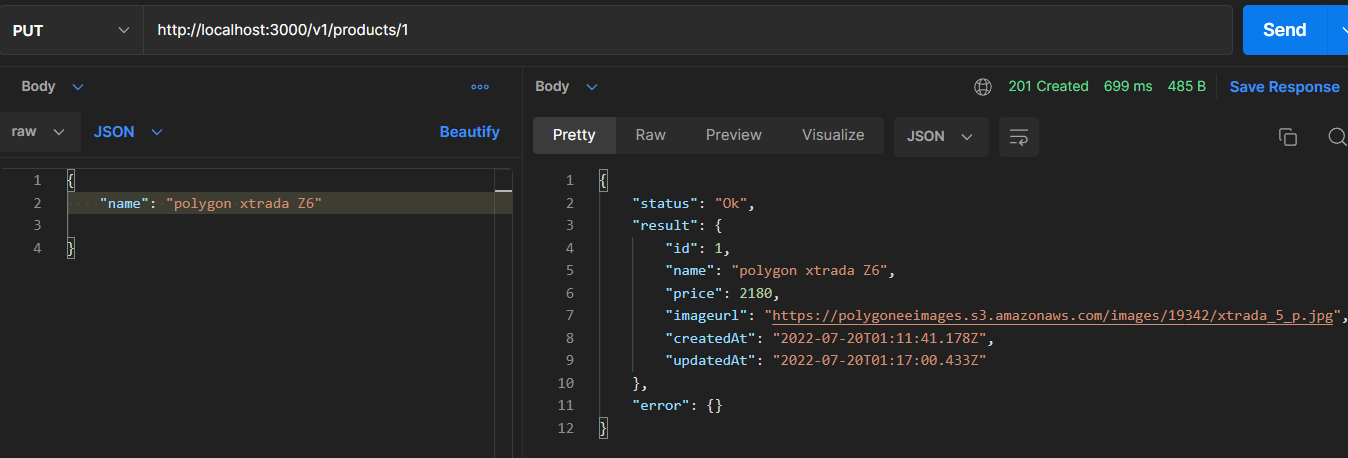
* Show Data by ID



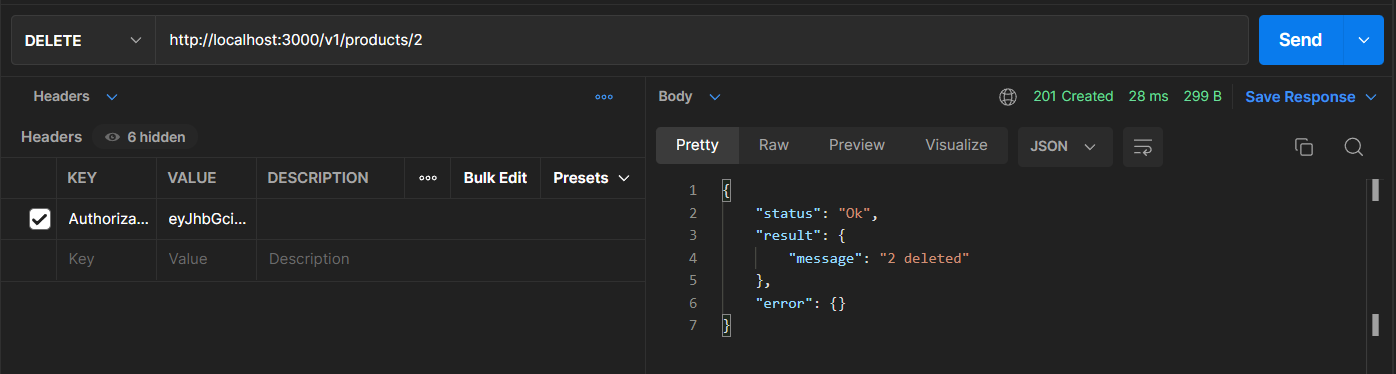
* Create Data



* Update Data



* Delete Data by ID



1. Dari dokumen <https://testbinar.docs.apiary.io/>, menurut anda, apakah ada desian API yang kurang maupun keliru? Jika ada, tuliskan kekurangan-kekurangan desain tersebut dan bagaimana seharusnya dokumentasi itu ditulis.

Jawab :

untuk dokumen https://testbinar.docs.apiary.io/ menurut saya mudah dipahami penggunaannya seperti apa.

tapi mungkin bisa ditambahkan dokumentasi request yang error karena frond end akan bingung bila terjadi error dalam sistem mereka dan dari backend tidak ada handling error.

Dan untuk update product sebaiknya seluruh data di ikut sertakan walaupun ada data yang tidak berubah

Jadi menurut saya saat membuat sebuah dokumentasi API alangkah baik nya dipisahkan yang request success dan request error. Contoh request success http status 200 sedangkan request error http status 500