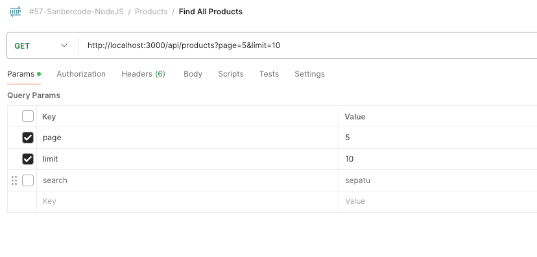
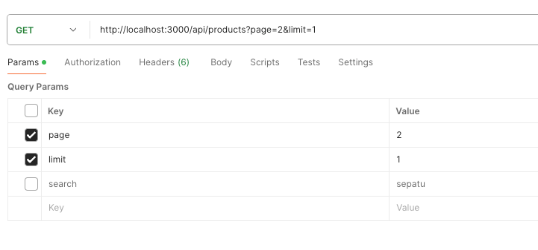
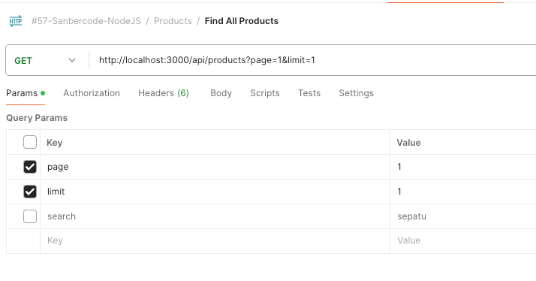
**Tugas 11: Validation, Mongoose Middleware dan Pagination Query**

NodeJS Backend Development with Xpress (Batch 62)

Nama: Amanatulhay Pribadi

**Latihan**

1. Lakukan request terhadap endpoint GET /products dengan parameter query seperti ini

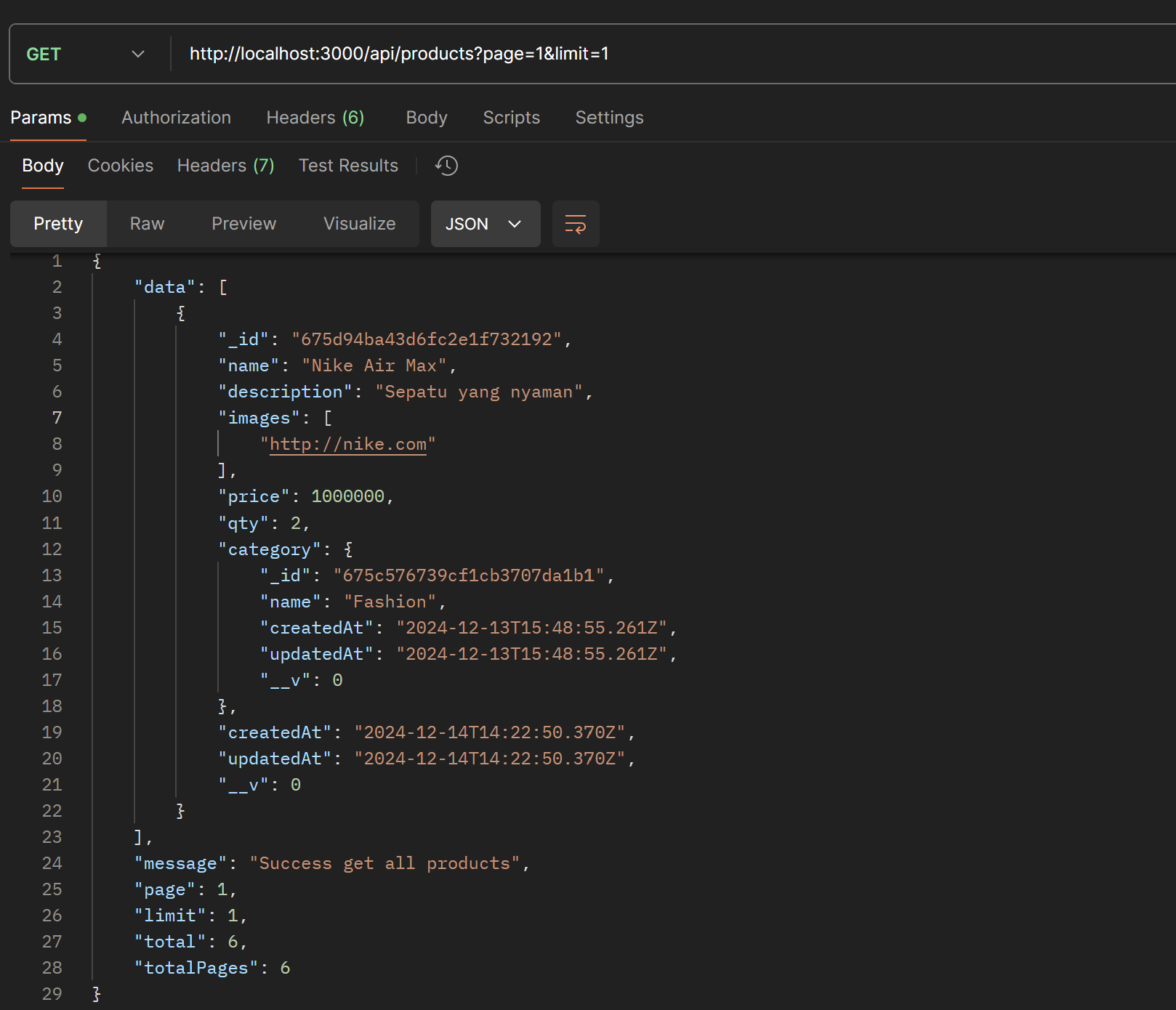


*Jawaban:*

Soal 1 kali ini berisi tiga request yang mengeksplorasi variasi query: *page* dan *limit*. Pada database terdapat 6 data products.

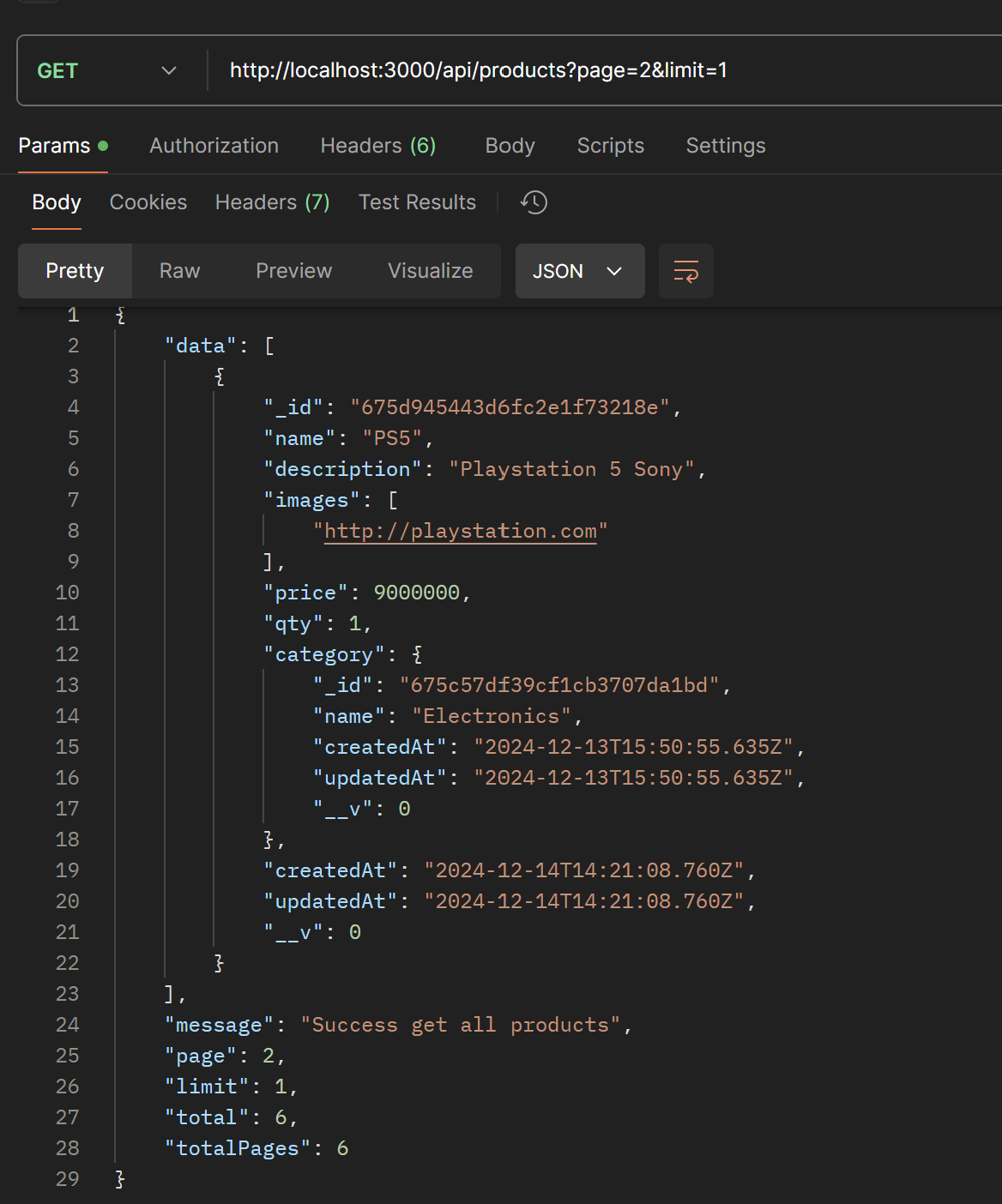
1. <http://localhost:3000/api/products?page=1&limit=1>

Karena setiap halaman dibatasi 1 data dokumen (*limit=1*), maka terdapat 6 halaman. Request meminta page ke-1 (*page=1*), karena itu data yang ditampilkan adalah urutan pertama.



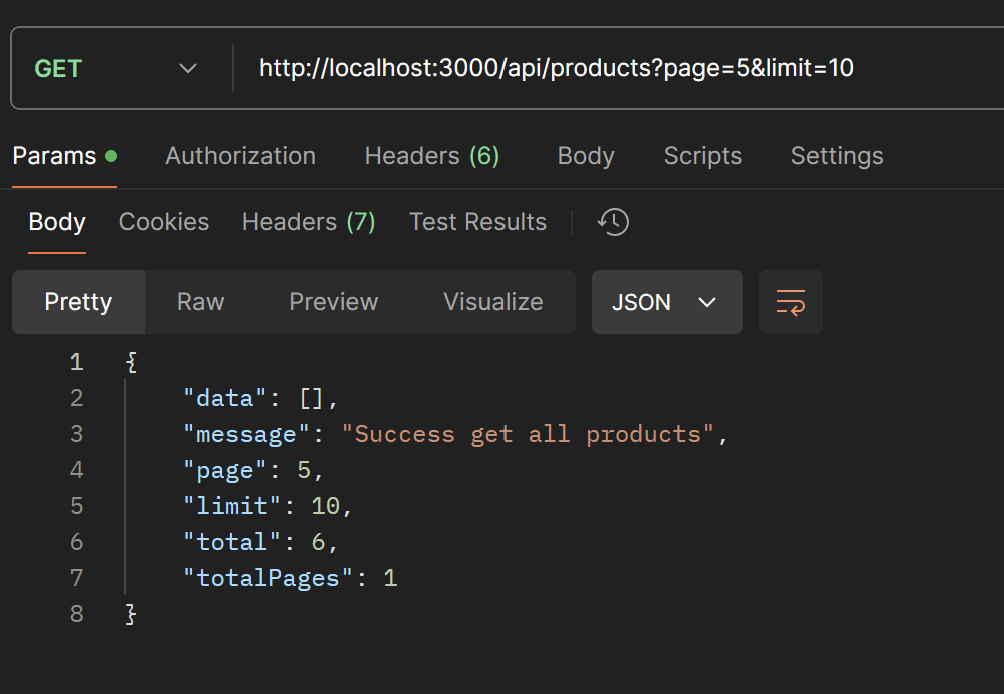
1. <http://localhost:3000/api/products?page=2&limit=1>

Sama seperti soal sebelumnya (*limit=1*), maka terdapat 6 halaman. Request meminta page ke-2 (*page=2*), karena itu data yang ditampilkan adalah data kedua.

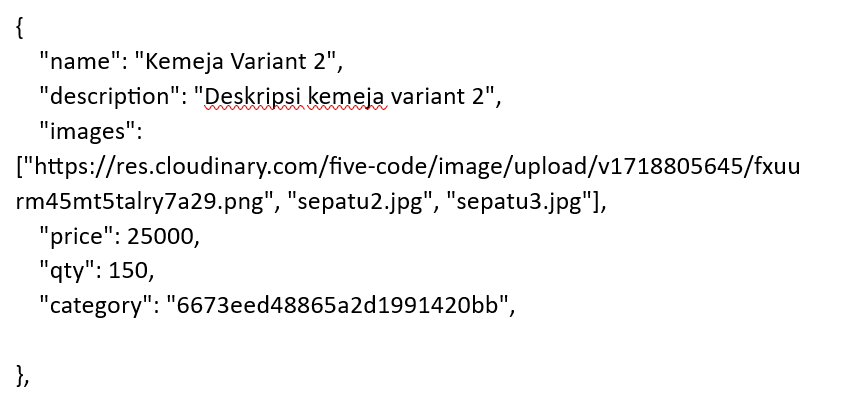
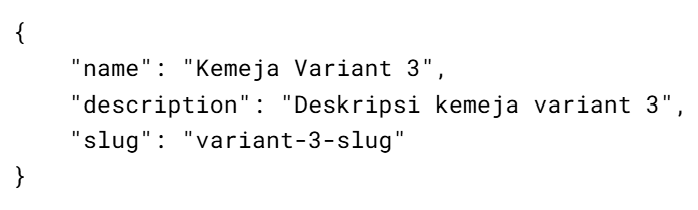


1. <http://localhost:3000/api/products?page=5&limit=10>

Karena setiap halaman dibatasi 10 data dokumen (*limit=10*) sementara jumlah data hanya 6, maka hanya terdapat 1 halaman yang berisi data. Request meminta page ke-5 (*page=5*) yang mana kosong.

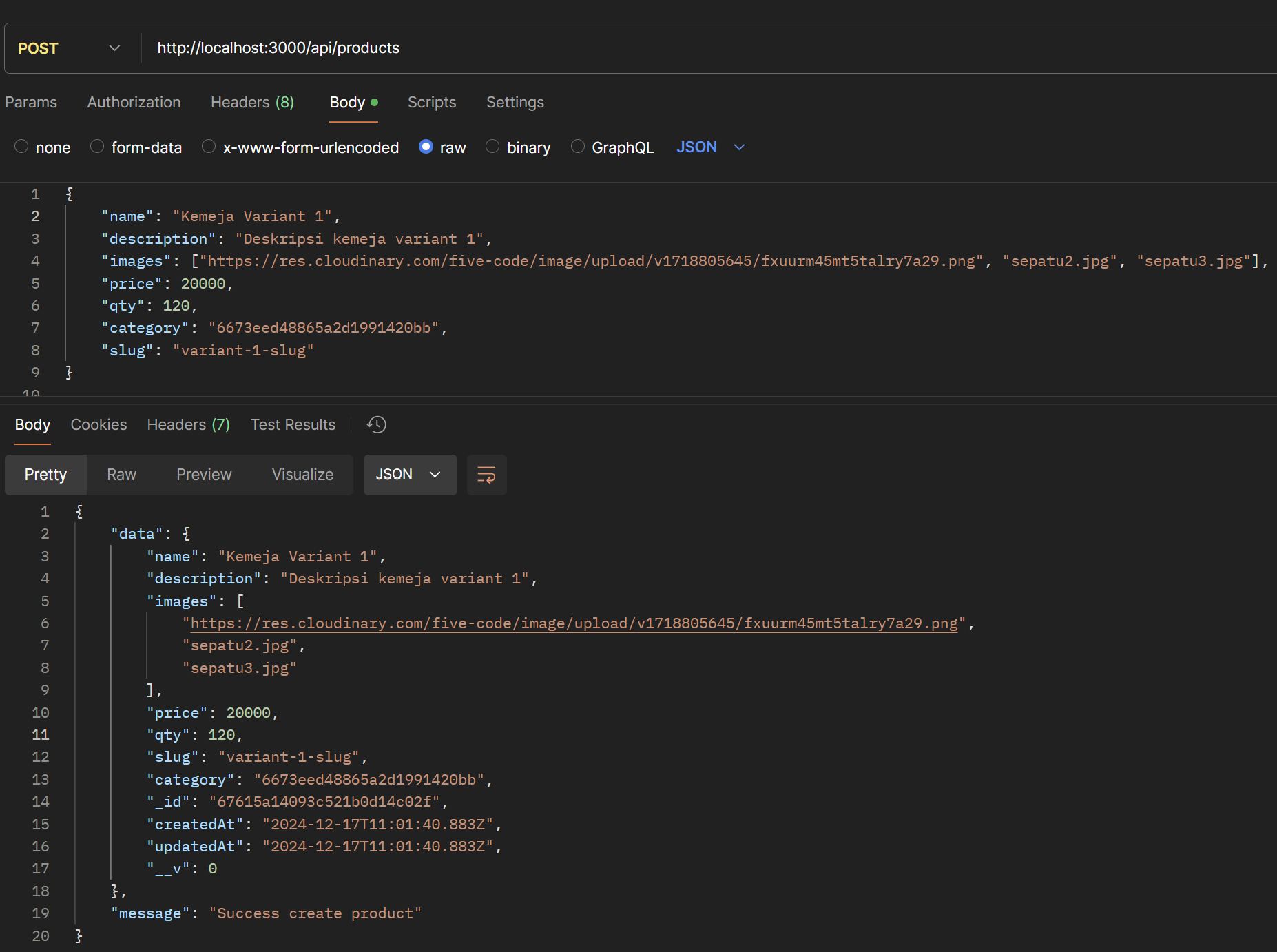


1. Lakukan request terhadap endpoint POST /products dengan request body seperti ini

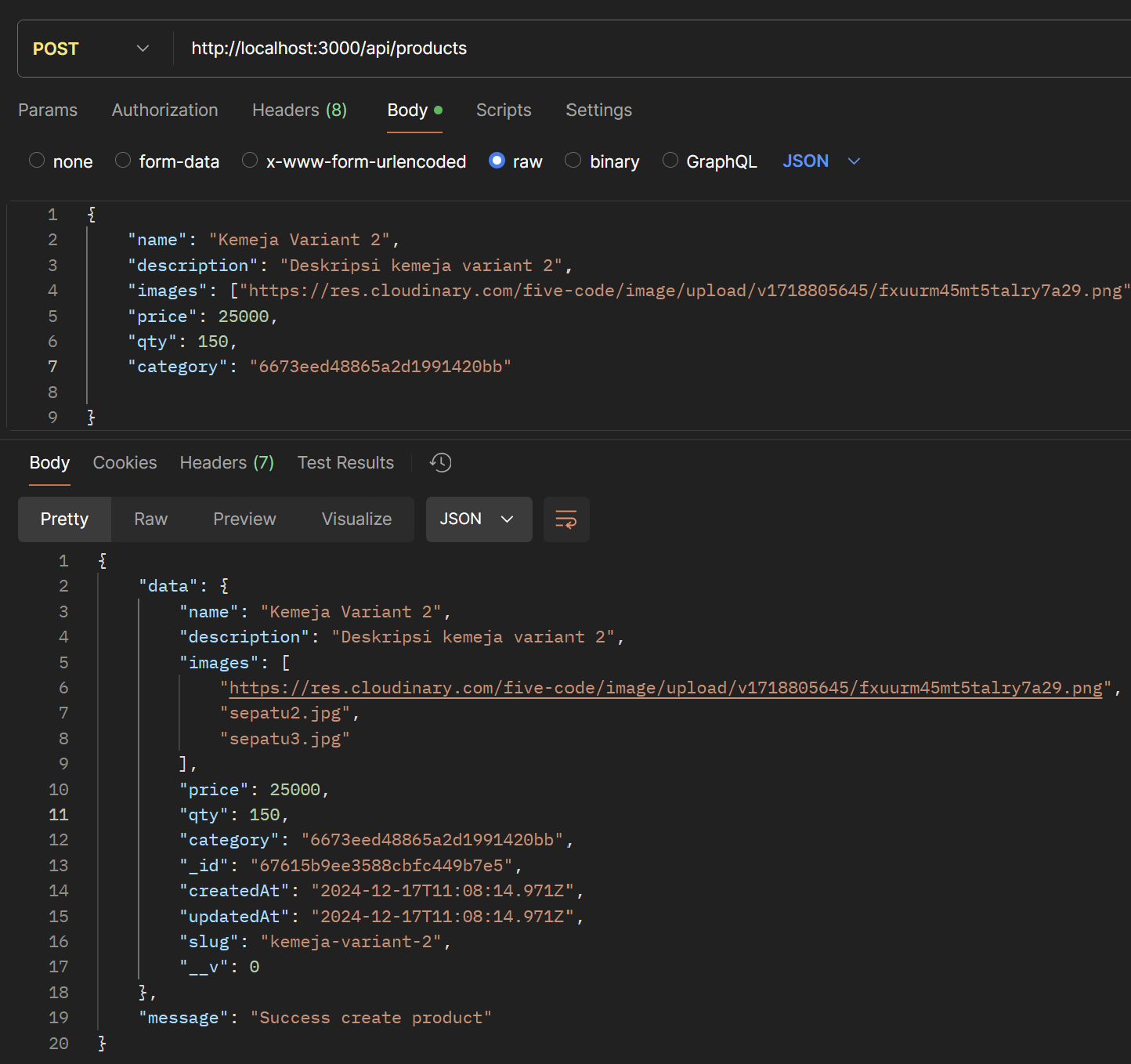
  

*Jawaban:*

Pada soal ini, kita akan meng-*explore* POST /products dengan variasi opsional properti *slug* pada Request body.



1. Pada request pertama, input slug ada. Karena itu, create product akan berhasil dan memiliki field slug yang berisi “variant-1-slug” sesuai Body Request bagian slug, seperti yang ditampilkan gambar di atas.

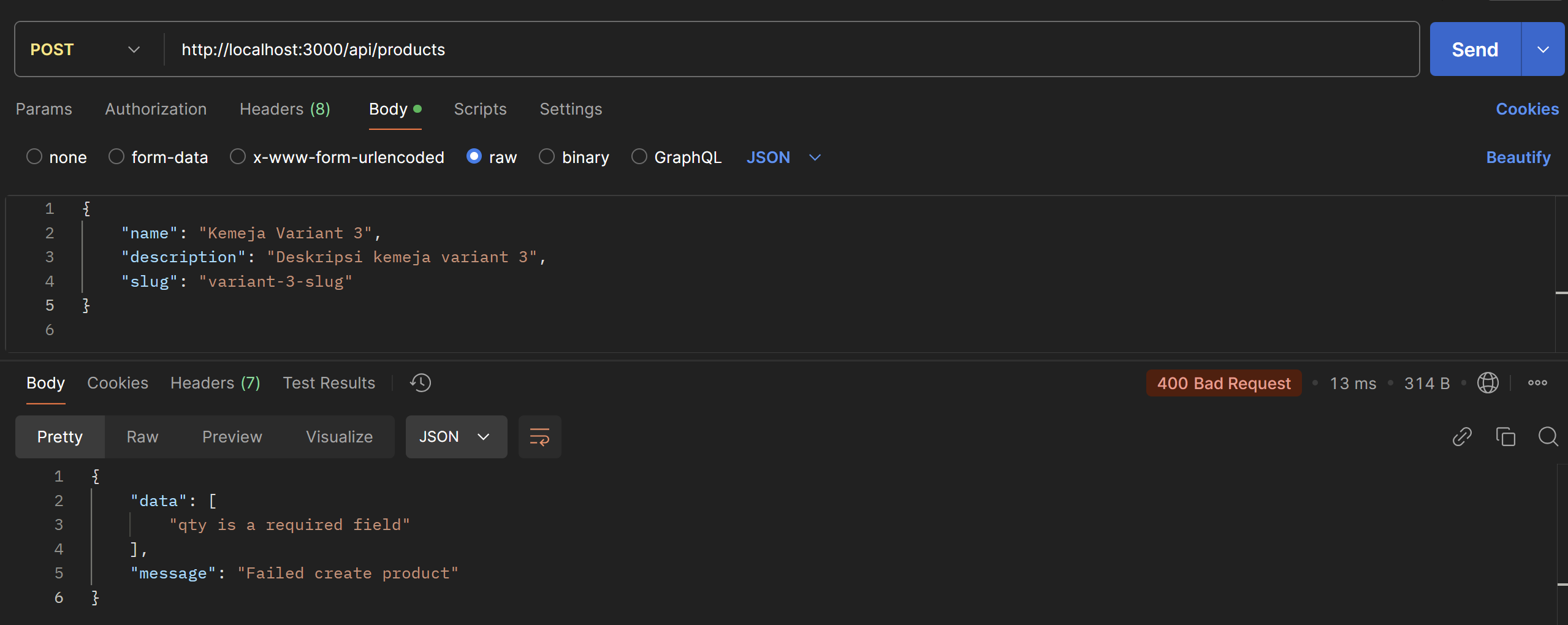


1. Gambar di atas menunjukkan request kedua di mana request Body tidak memiliki data *slug*. Namun karena pada schema telah ditambah kode di bawah ini:

*product.slug = product.name.toLowerCase().split(" ").join("-");*

Maka, create data product akan menggunakan value pada nama yaitu Kemeja Variant 2 yang dikecilkan, tanpa spasi, dan diberi karakter sambung “-“.

Terlihat pada gambar di atas bahwa slug bernilai “kemeja-variant-2” sesuai nama product.



1. Gambar di atas memperlihatkan hasil Request dengan error code 400 Bad Request, dikarenakan Body yang tidak lengkap. Karena pada schema validation salah satunya tertulis:

*qty: Yup.number().required().min(1),*

Maka variable qty menjadi persyaratan/required sehingga request tidak berhasil.

Ringkasan

Pada materi dan tugas hari ini dapat diambil beberapa pelajaran berikut.

1. Yup dapat digunakan untuk validasi data dengan schema yang dirancang. Schema ini juga dapat berisi persyaratan required atau tipe data seperti number atau string.
2. Function create product yang ditambahkan validasi telah berhasil menyeleksi inputan request yang tidak sesuai aturan skema, misalkan, qty product bernilai kosong.
3. Mongoose middleware dapat menambahkan aksi yang dijalankan sebelum atau sesudah operasi database, seperti save. Pada Latihan kita telah mencoba untuk membuat middleware pre(‘save’) yang mana mengisi data slug apabila kosong berdasarkan nama product.
4. Untuk memudahkan pengambilan data yang lebih terorganisir, kita dapat menggunakan fitur Pagination. Kita dapat mengatur sehingga tampilan halaman data berdasarkan nilai page, limit, dan search yang dimasukkan oleh user. Pagination sangat penting apabila kita bekerja dengan data yang sangat banyak, sehingga user butuh tampilan data yang lebih tertata.