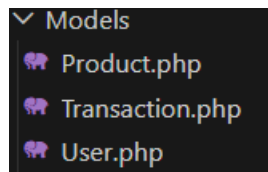
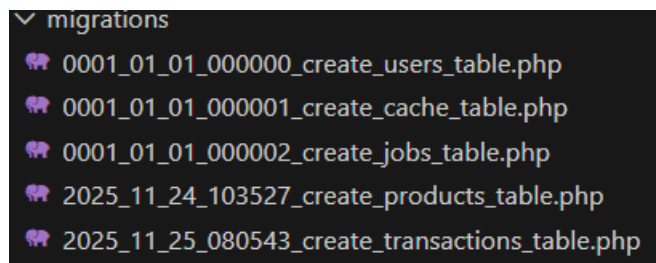


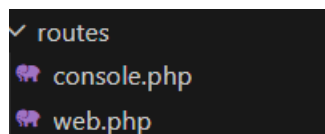
Pada controller di sini ini kita hanya menggunakna 2 yaitu controller untuk produk dan transaction



Pada model di sini ini kita hanya menggunakna 2 yaitu model untuk produk dan transaction

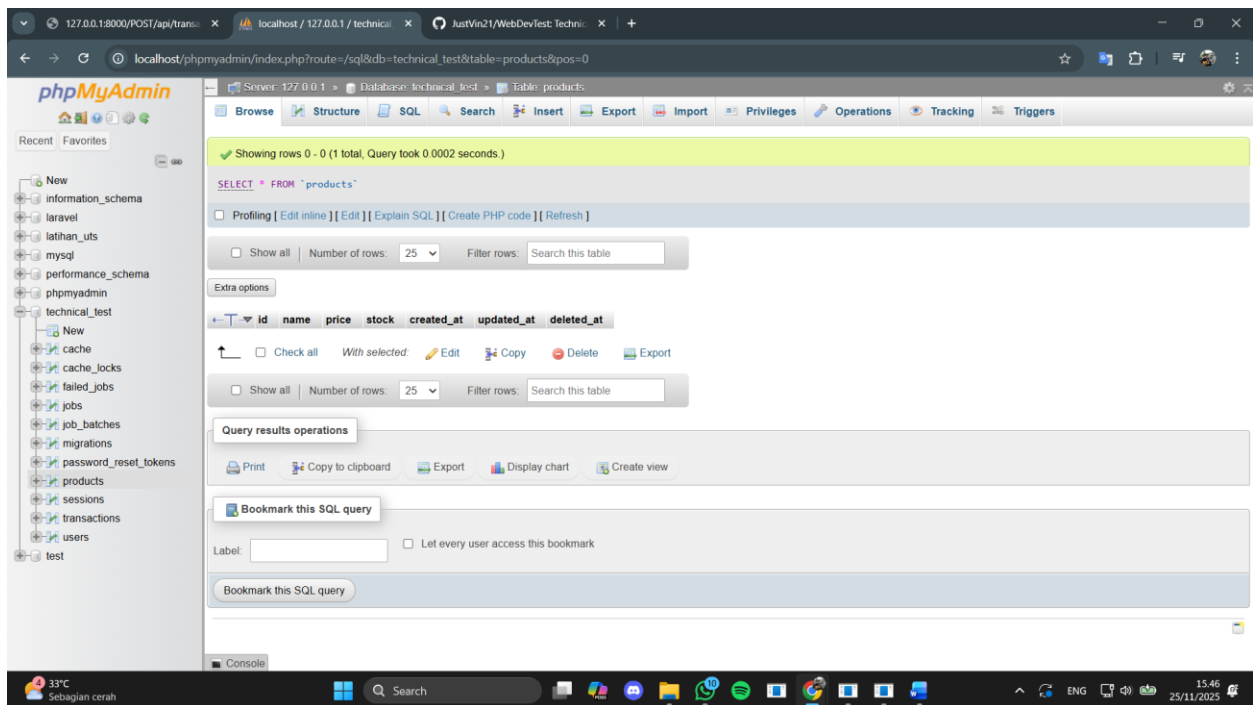


Pada migration juga kita menggunkan 2 yaitu migration untuk table product dan transaction

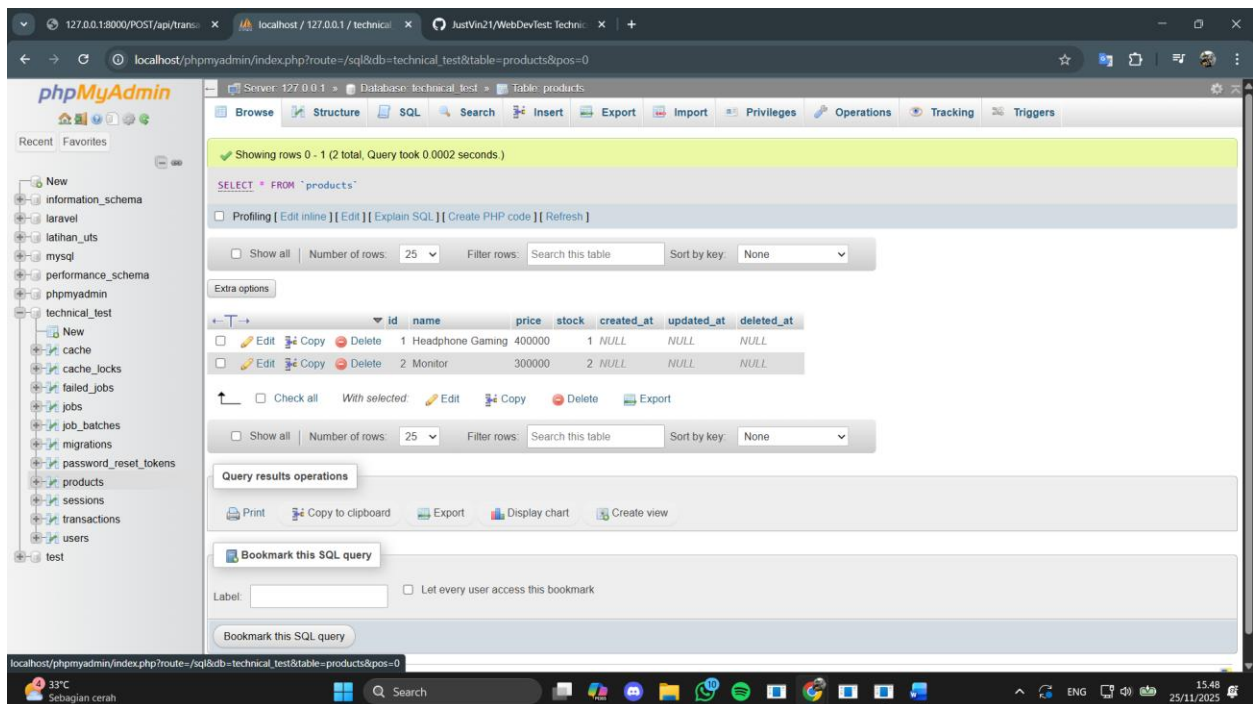


Untuk routing saya guanakan dan masukkan codenya pada web.php karena code yang di buat pada route tidak banyak sehingga menurut saya akan lebih baik jika hanya pada 1 file.

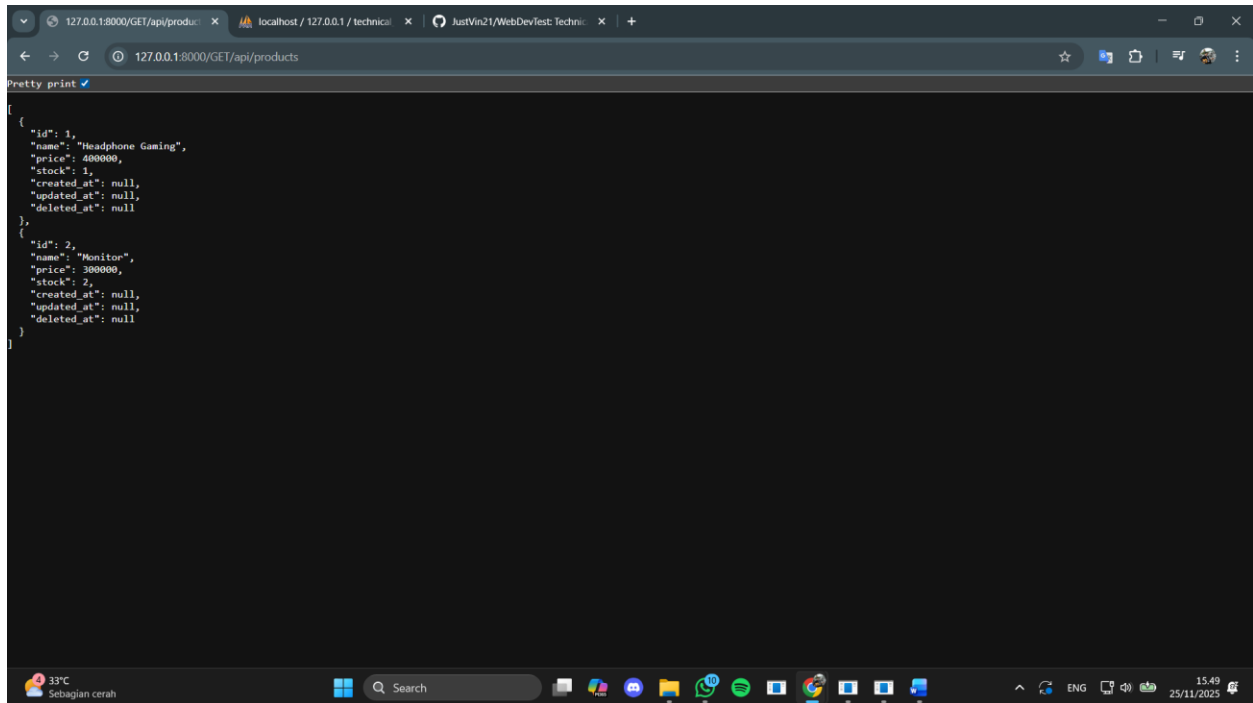
Sekarang kita akan masuk ke dalam test back endnya



Disini saya menggunakan mysql pada XAMPP, disini saya akan memasukkan data dummy pada table product.



Disini saya telah membuat 2 data dummy dengan id 1 “Headphone Gaming” dan id 2 “Monitor”

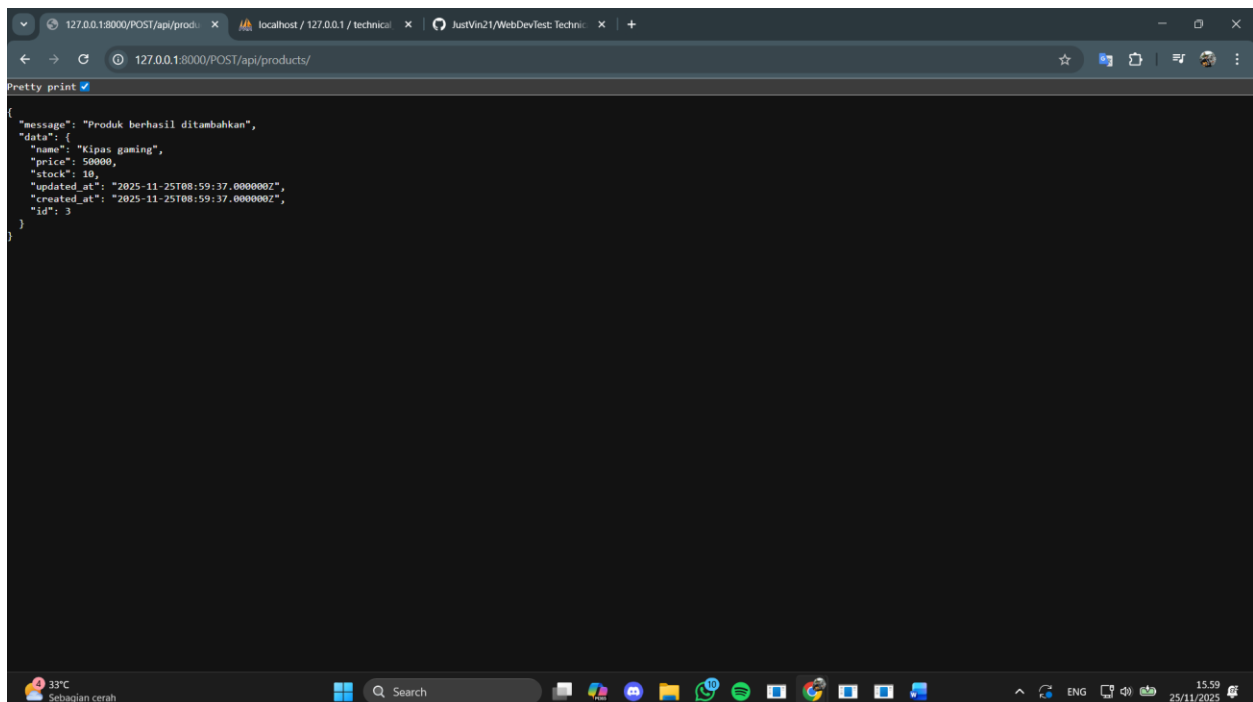


```
127.0.0.1:8000/GET/api/products

Pretty print

[
  {
    "id": 1,
    "name": "Headphone Gaming",
    "price": 400000,
    "stock": 1,
    "created_at": null,
    "updated_at": null,
    "deleted_at": null
  },
  {
    "id": 2,
    "name": "Monitor",
    "price": 300000,
    "stock": 2,
    "created_at": null,
    "updated_at": null,
    "deleted_at": null
  }
]
```

Sekarang kita akan lihat data yang telah di masukkan tadi. Dengan memasukkan link <http://127.0.0.1:8000/GET/api/products> yang bertujuan untuk GET (melihat semua data pada table) dapat kita lihat bahwa 2 data dummy yang dimasukkan pada mysql tadi telah muncul dan dapat dilihat.

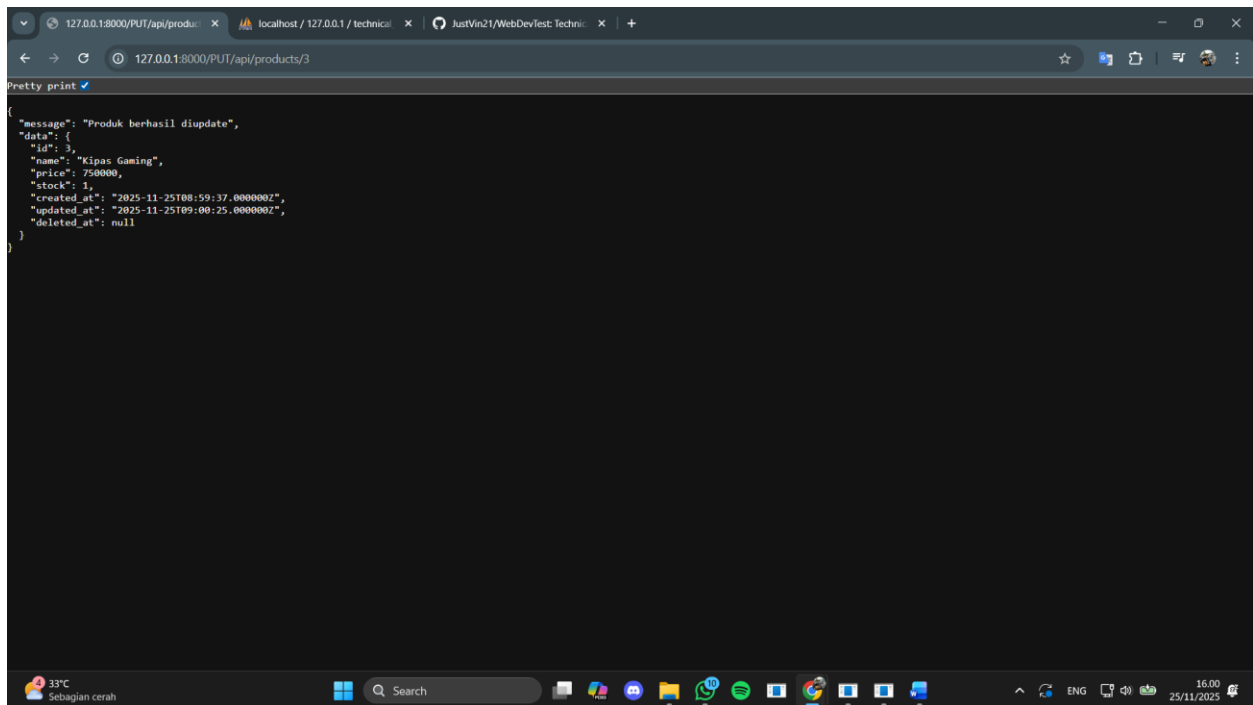


```
127.0.0.1:8000/POST/api/products/

Pretty print

{
  "message": "Produk berhasil ditambahkan",
  "data": {
    "name": "Kipas gaming",
    "price": 50000,
    "stock": 10,
    "updated_at": "2025-11-25T08:59:37.000000Z",
    "created_at": "2025-11-25T08:59:37.000000Z",
    "id": 3
  }
}
```

Disini kita akan melakukan POST yang bertujuan untuk menambahkan data dummy namun tidak melalui mysql tetapi bisa secara langsung pada Laravel. Dengan memasukkan link <http://127.0.0.1:8000/POST/api/products> maka kita dapat melakukan POST. Setelah melakukan POST maka data akan masuk ke table dan di simpan.

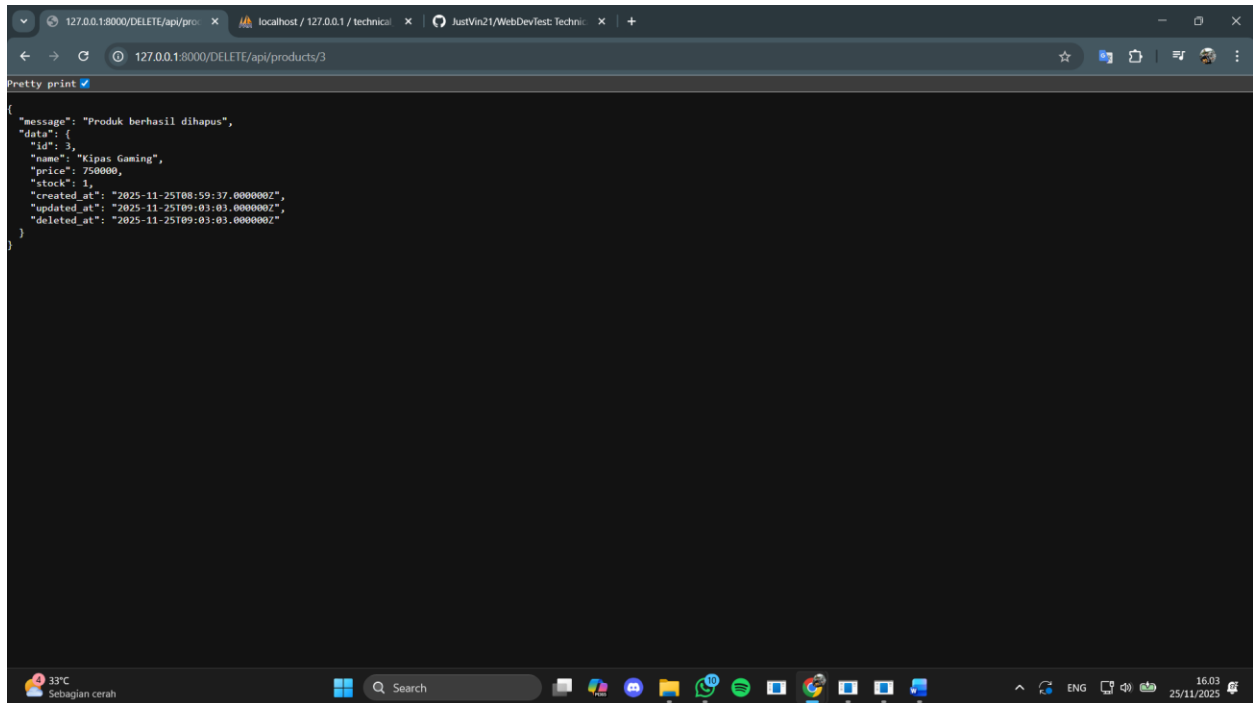


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/PUT/api/products/3`. The main content area shows a JSON response from a REST client, labeled "Pretty print". The response is as follows:

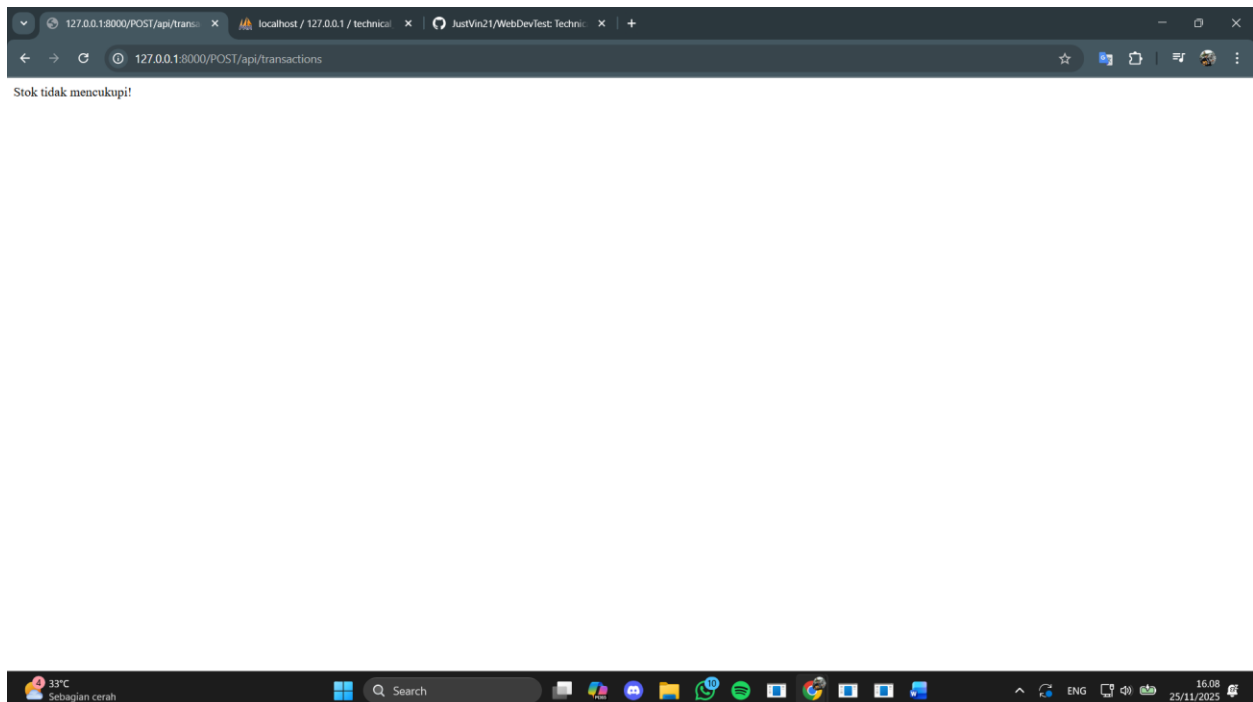
```
{
  "message": "Produk berhasil diupdate",
  "data": {
    "id": 3,
    "name": "Kipas Gaming",
    "price": 750000,
    "stock": 3,
    "created_at": "2025-11-25T08:59:37.000000Z",
    "updated_at": "2025-11-25T09:00:25.000000Z",
    "deleted_at": null
  }
}
```

The browser's taskbar at the bottom shows the system clock as 16:00 on 25/11/2023.

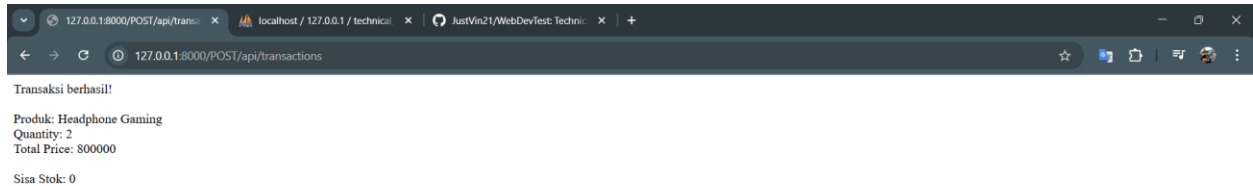
Disini kita melakukan PUT yang bertujuan untuk mengupdate data lama ke data yang terbaru. Dengan memasukkan link ini <http://127.0.0.1:8000/PUT/api/products/3> maka akan mengupdate data pada id 3 dengan data yang terbaru. Data akan di simpan di database.



Disini kita melakukan DELETE yang bertujuan untuk menghapus data yang ada. Dengan memasukkan link <http://127.0.0.1:8000/DELETE/api/products/3> maka akan secara otomatis menghapus data dengan id 3 dan akan hilang dari data base pada Laravel namun tidak pada database mysql jadinya seperti di hide dari Laravel namun masih ada pada mysql.



Disini kita melakukan POST transactions dengan id 1 “Headphone Gaming” karena kita akan melakukan pembelian id 1 dengan quantity 2 maka headphone gaming tidak dapat di beli karena jumlah stock kurang. Jika stock kurang maka akan menampilkan pesan stock tidak mencukupi.



Namun ketika kita menyiapkan stocknya sama dengan atau lebih dari quantity yang di beli maka product dapat di beli dan akan menampilkan pesan ini. Dengan memasukkan link <http://127.0.0.1:8000/POST/api/transactions> maka kita dapat membeli product dan akan mengupdate data terbaru.