

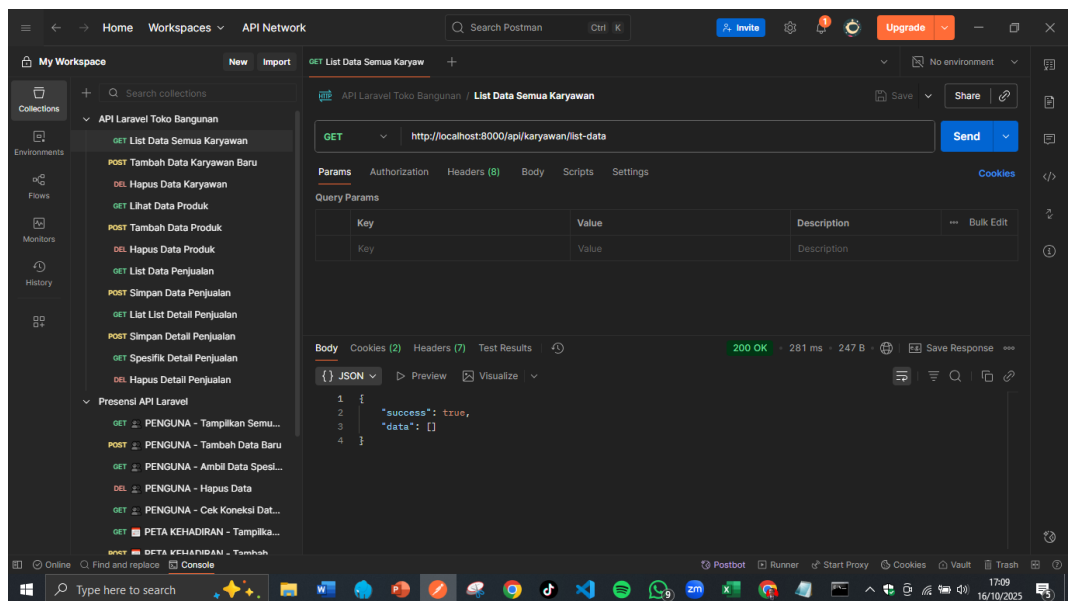
Nama : Stevanus Andika Galih Setiawan.

Kelas : RK231.

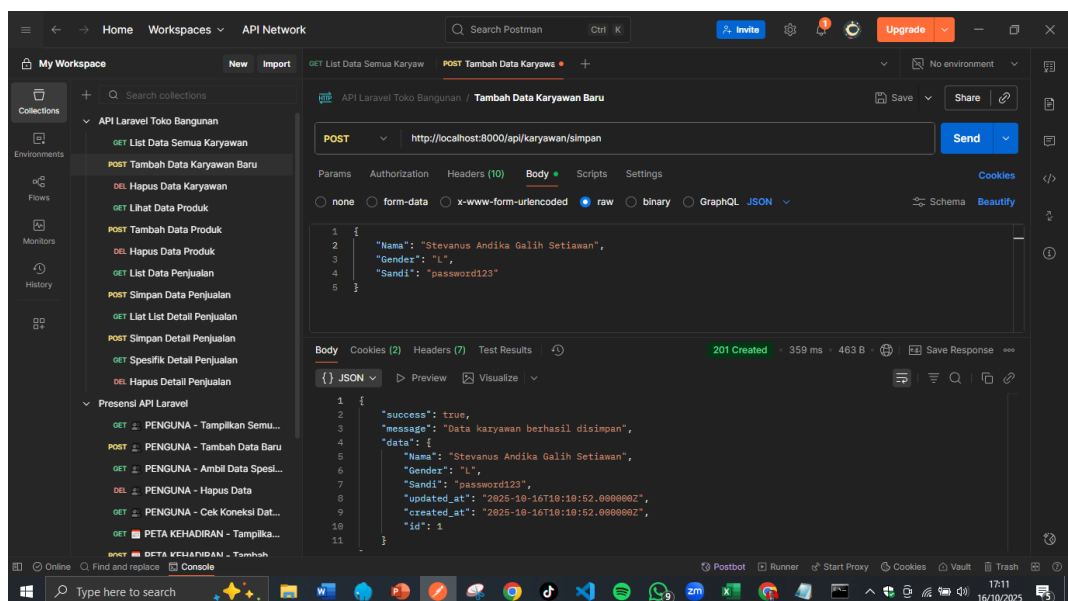
NIM : 202303110008.

Tugas API Toko Bangunan

1. List Tampilkan Data Karyawan:



2. Tambah Data Karyawan.



3. Hapus Data Karyawan

The image displays two screenshots of the Postman API client interface, showing the configuration and results of API requests.

Top Screenshot: DELETE Request

- Method:** DELETE
- URL:** `http://localhost:8000/api/karyawan/hapus/1`
- Status:** 200 OK
- Response Body (JSON):**

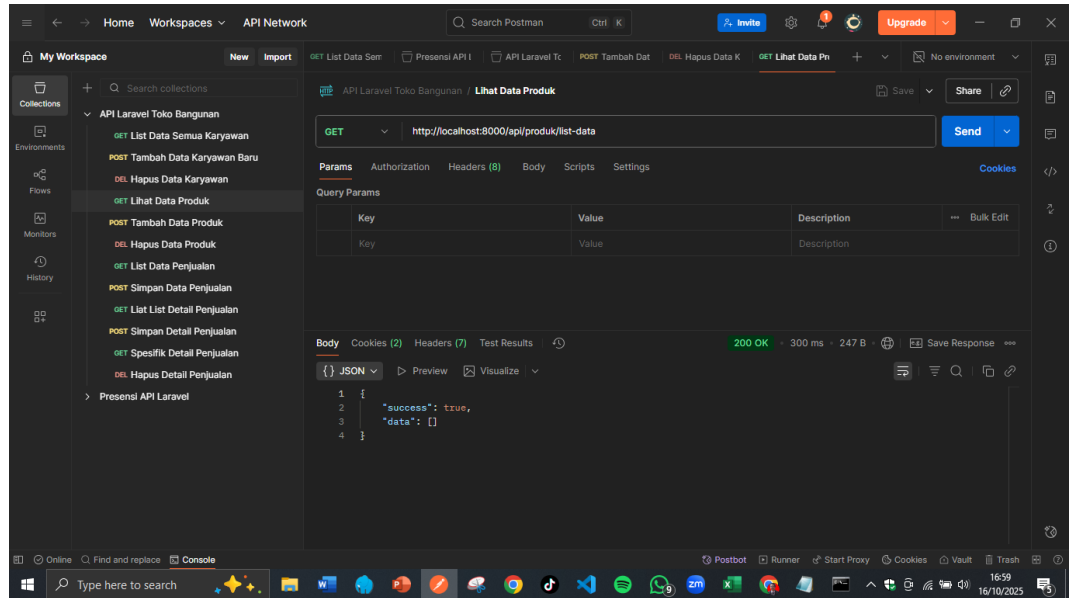
```
{  "success": true,  "message": "Data karyawan berhasil dihapus"}
```

Bottom Screenshot: GET Request

- Method:** GET
- URL:** `http://localhost:8000/api/karyawan/list-data`
- Status:** 200 OK
- Response Body (JSON):**

```
{  "success": true,  "data": []}
```

4. Tampilkan List Data Produk



5. Tambah Produk

The image displays two screenshots of the Postman API client interface, showing the 'Tambah Data Produk' endpoint being tested. Both screenshots show a successful POST request to the URL `http://localhost:8000/api/produk/simpan` with a status of 201 Created.

First Screenshot (Top):

- Request Body (raw):**

```
1 {
2   "Produk": "Cat Dinding Nippon Paint",
3   "Harga": 58000,
4   "Stok": 15
5 }
```
- Response Body (JSON):**

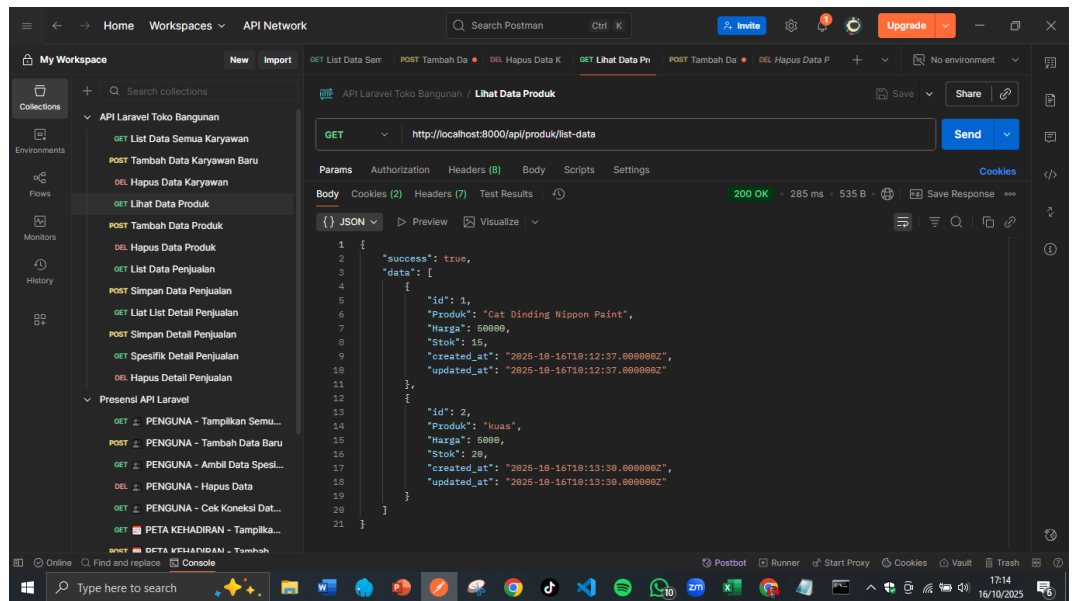
```
1 {
2   "success": true,
3   "message": "Data produk berhasil disimpan",
4   "data": {
5     "Produk": "Cat Dinding Nippon Paint",
6     "Harga": 58000,
7     "Stok": 15,
8     "updated_at": "2025-10-16T10:12:37.000000Z",
9     "created_at": "2025-10-16T10:12:37.000000Z",
10    "id": 1
11  }
12 }
```

Second Screenshot (Bottom):

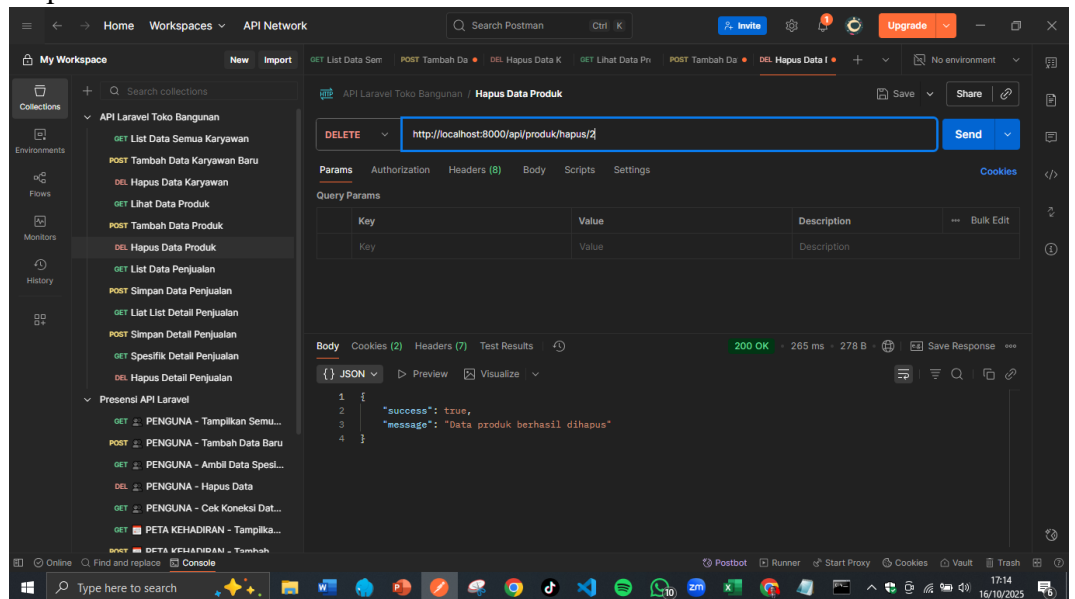
- Request Body (raw):**

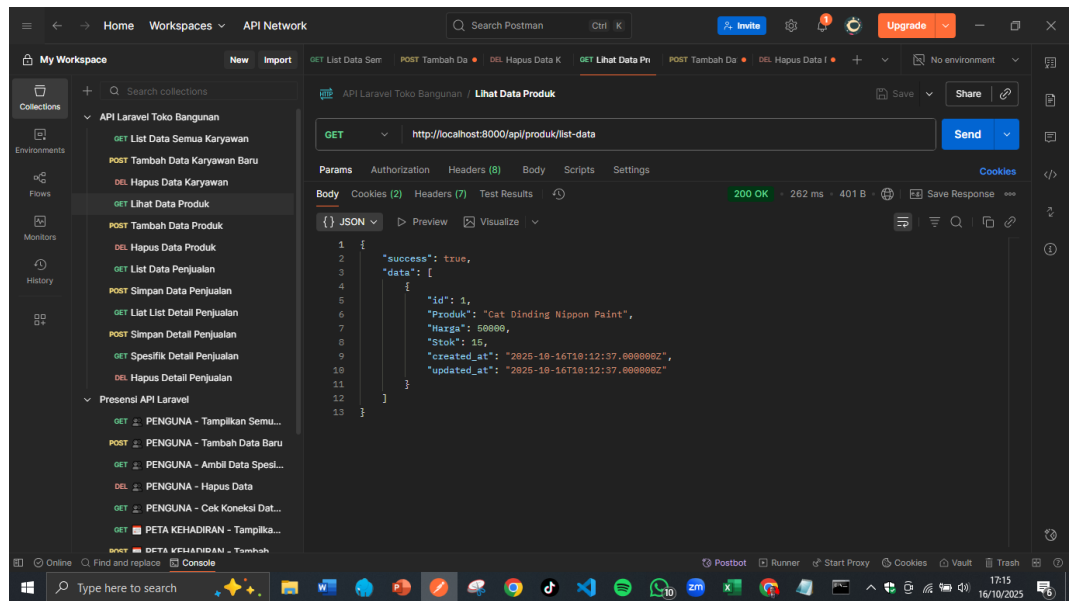
```
1 {
2   "Produk": "kuas",
3   "Harga": 5800,
4   "Stok": 20
5 }
```
- Response Body (JSON):**

```
1 {
2   "success": true,
3   "message": "Data produk berhasil disimpan",
4   "data": {
5     "Produk": "kuas",
6     "Harga": 5800,
7     "Stok": 20,
8     "updated_at": "2025-10-16T10:13:30.000000Z",
9     "created_at": "2025-10-16T10:13:30.000000Z",
10    "id": 2
11  }
12 }
```

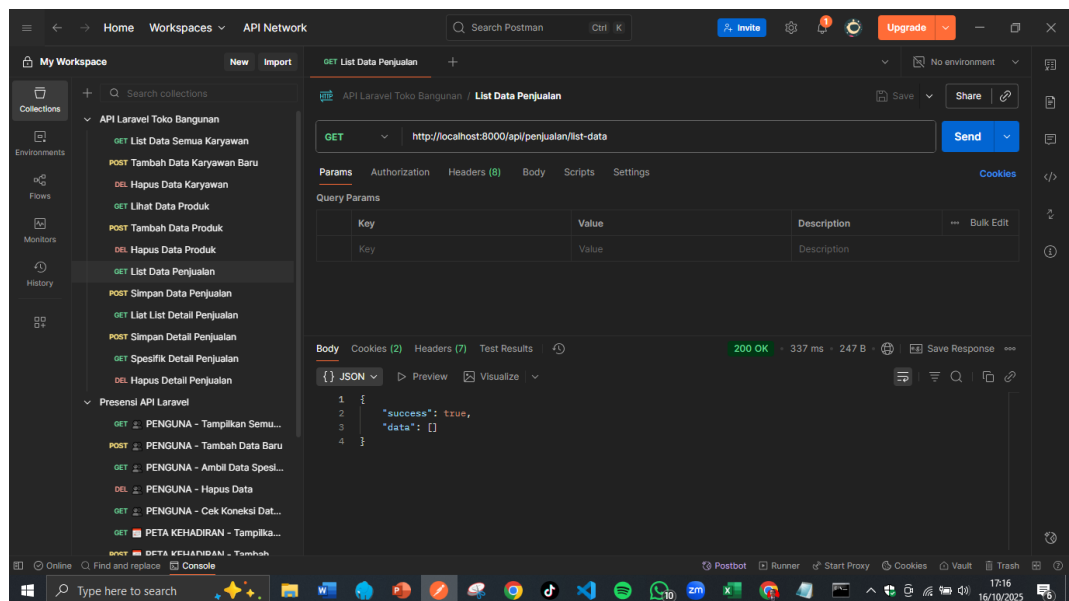


6. Hapus Produk

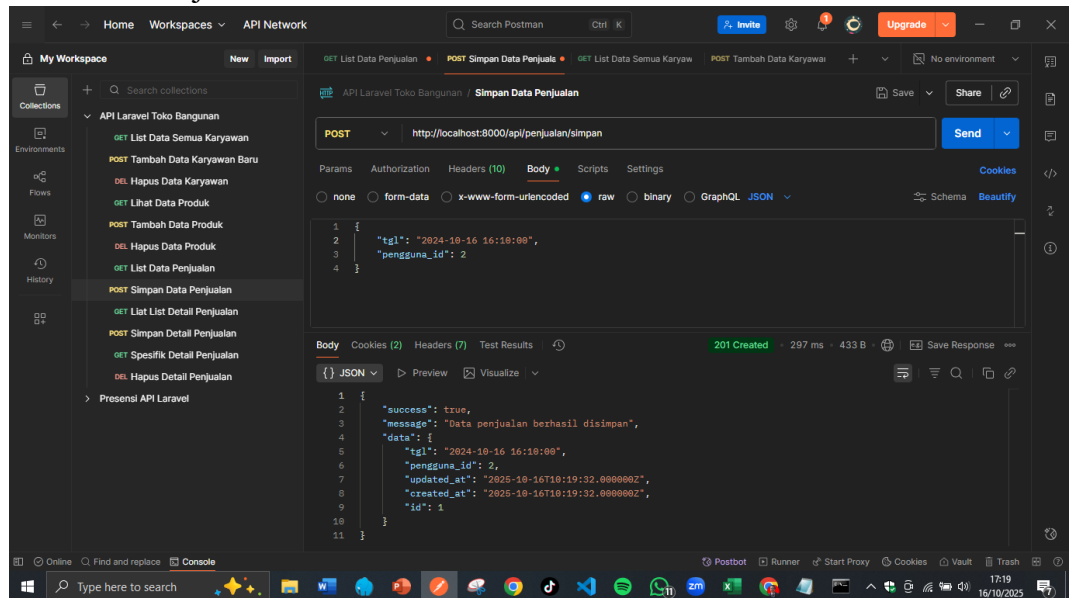




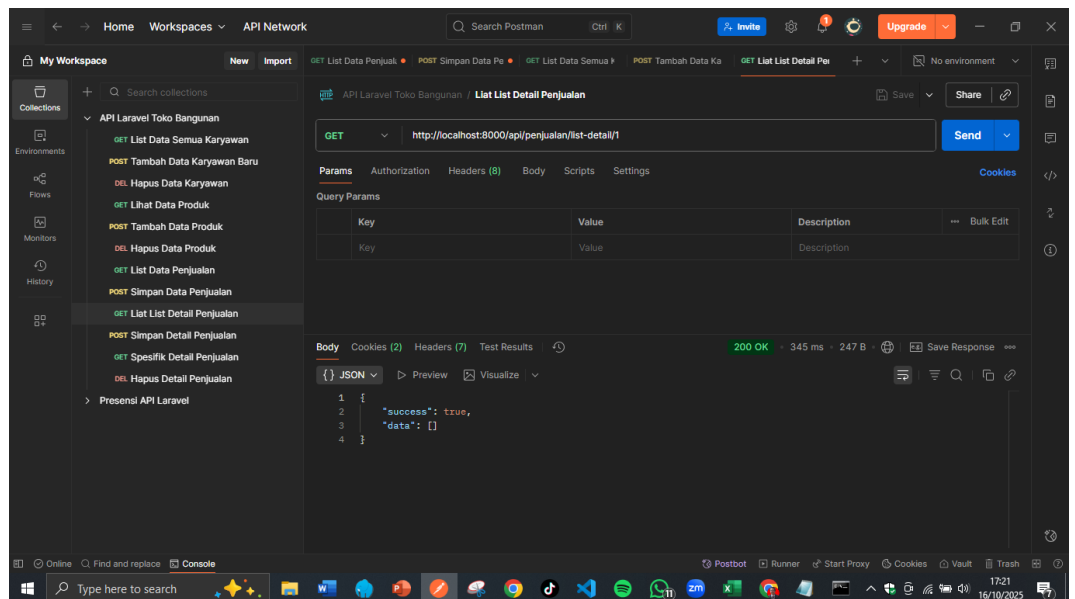
7. List Data Penjualan



8. Buat Data Penjualan



9. List Detail Penjualan



10. Tambah Detail Penjualan

The image displays two screenshots of the Postman API client interface, illustrating the process of adding a sales detail.

Top Screenshot: POST Request

- Collection:** API Laravel Toko Bangunan
- Environment:** API Laravel Toko Bangunan
- Request:** POST `http://localhost:8000/api/penjualan/simpan-detail`
- Body (raw):**

```
1 {
2   "penjualan_id": 1,
3   "produk_id": 1,
4   "qty": 2,
5   "harga": 100000
6 }
```
- Response (JSON):**

```
1 {
2   "success": true,
3   "message": "Detail penjualan berhasil disimpan"
4 }
```

Bottom Screenshot: GET Request

- Collection:** API Laravel Toko Bangunan
- Environment:** API Laravel Toko Bangunan
- Request:** GET `http://localhost:8000/api/penjualan/list-detail/1`
- Response (HTML):**

```
1 [{"success":true,"data":[{"id":1,"penjualan_id":1,"produk_id":1,"qty":2,"harga":100000,
2   "created_at":"2025-10-16T18:22:34.000000Z","updated_at":"2025-10-16T18:22:34.000000Z","produk":{"id":1,
3     "Produk":"Cat
4     Dinding Nippon
5     Paint","Harga":50000,"Stok":13,"created_at":"2025-10-16T18:12:37.000000Z","updated_at":"2025-10-16T18:22:34.
6     000000Z"},"penjualan":{"id":1,"tgl":"2024-10-16
7     16:10:00","pengguna_id":2,"created_at":"2025-10-16T18:19:32.000000Z","updated_at":"2025-10-16T18:19:32.
8     000000Z"}}]}]
```


11. Hapus Detail Penjualan

