**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Ditujukan sebagai salah satu syarat untuk memperoleh nilai \_\_\_\_\_\_

pada mata kuliah Programming III (Webservice)

Dosen Pengampu : Indra Riksa Herlambang, S.Tr. Kom. M.Kom.

****

Disusun Oleh:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kelas 2 C

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**2022**

DAFTAR ISI

[**DAFTAR ISI** 2](#_Toc110521285)

[**1.** **Diagram ERD** 6](#_Toc110521286)

[1.1 Tabel Golongan 6](#_Toc110521287)

[1.2 Tabel Obat 7](#_Toc110521288)

[1.3 Tabel Transaksi Penjualan 7](#_Toc110521289)

[1.4 Tabel Transaksi Supplier 8](#_Toc110521290)

[1.5 Tabel Karyawan 9](#_Toc110521291)

[1.6 Tabel Pelanggan 9](#_Toc110521292)

[1.7 Tabel Supplier 10](#_Toc110521293)

[1.8 Tabel Keys 10](#_Toc110521294)

[1.9 Tabel Limits 11](#_Toc110521295)

[**2.** **Testing Masing-Masing Endpoint Pada Rest Server Menggunakan POSTMAN** 11](#_Toc110521296)

[2.1 Endpoint dan Hasil Testing pada Rest Server Obat 11](#_Toc110521297)

[2.1.1 Hasil Testing pada GET Obat 11](#_Toc110521298)

[2.1.2 Hasil Testing pada Update data Obat 12](#_Toc110521299)

[2.1.3 Hasil Testing pada Create data Obat 12](#_Toc110521300)

[2.1.4 Hasil Testing pada Delete data Obat 13](#_Toc110521301)

[2.2 Endpoint dan Hasil Testing pada Rest Server Golongan Obat 13](#_Toc110521302)

[2.2.1 Hasil Testing pada GET Golongan Obat 13](#_Toc110521303)

[2.2.2 Hasil Testing pada Update Golongan Obat 14](#_Toc110521304)

[2.2.3 Hasil Testing pada Create Golongan Obat 15](#_Toc110521305)

[2.2.4 Hasil Testing pada Delete Golongan Obat 15](#_Toc110521306)

[2.3 Endpoint dan Hasil Testing pada Rest Server Karyawan 16](#_Toc110521307)

[2.3.1 Hasil Testing pada GET Karyawan 16](#_Toc110521308)

[2.3.2 Hasil Testing pada Update Karyawan 16](#_Toc110521309)

[2.3.3 Hasil Testing pada Create Karyawan 17](#_Toc110521310)

[2.3.4 Hasil Testing pada Delete Karyawan 18](#_Toc110521311)

[2.4 Endpoint dan Hasil Testing pada Rest Server Supplier 18](#_Toc110521312)

[2.4.1 Hasil Testing pada GET Supplier 18](#_Toc110521313)

[2.4.2 Hasil Testing pada Update Supplier 19](#_Toc110521314)

[2.4.3 Hasil Testing pada Create Supplier 19](#_Toc110521315)

[2.4.4 Hasil Testing pada Delete Supplier 20](#_Toc110521316)

[2.5 Endpoint dan Hasil Testing pada Rest Server Pelanggan 21](#_Toc110521317)

[2.5.1 Hasil Testing pada GET Pelanggan 21](#_Toc110521318)

[2.5.2 Hasil Testing pada Update Pelanggan 21](#_Toc110521319)

[2.5.3 Hasil Testing pada Create Pelanggan 22](#_Toc110521320)

[2.5.4 Hasil Testing pada Delete Supplier 22](#_Toc110521321)

[2.6 Endpoint dan Hasil Testing pada Rest Server Transaksi Penjualan 23](#_Toc110521322)

[2.6.1 Hasil Testing pada GET Transaksi Penjualan 23](#_Toc110521323)

[2.6.2 Hasil Testing pada Update Transaksi Penjualan 23](#_Toc110521324)

[2.6.3 Hasil Testing pada Create 24](#_Toc110521325)

[2.6.4 Hasil Testing pada Delete Transaksi Penjualan 25](#_Toc110521326)

[2.7 Endpoint dan Hasil Testing pada Rest Server Transaksi Penjualan 25](#_Toc110521327)

[2.7.1 Hasil Testing pada GET Transaksi Penjualan 25](#_Toc110521328)

[2.7.2 Hasil Testing pada Update Transaksi Supplier 26](#_Toc110521329)

[2.7.3 Hasil Testing pada Create 26](#_Toc110521330)

[2.7.4 Hasil Testing pada Delete Transaksi Penjualan 27](#_Toc110521331)

[**3.** **Output untuk masing-masing endpoint** 28](#_Toc110521332)

# DAFTAR TABEL

[Tabel 1. 1 Tabel Golongan 6](#_Toc110521476)

[Tabel 1. 2 Tabel Obat 7](#_Toc110521477)

[Tabel 1. 3 Tabel Transaksi Penjualan 8](#_Toc110521478)

[Tabel 1. 4 Tabel Transaksi Supplier 8](#_Toc110521479)

[Tabel 1. 5 Tabel Karyawan 9](#_Toc110521480)

[Tabel 1. 6 Tabel Pelanggan 9](#_Toc110521481)

[Tabel 1. 7 Tabel Supplier 10](#_Toc110521482)

[Tabel 1. 8 Tabel Keys 10](#_Toc110521483)

[Tabel 1. 9 Tabel Limits 11](#_Toc110521484)

# DAFTAR GAMBAR

[Gambar 1. 1 Diagram ERD 6](#_Toc110521528)

[Gambar 2. 1 Hasil Testing pada GET Obat……………………………………………………...11](#_Toc110521683)

[Gambar 2. 2 Hasil Testing pada Update data Obat 12](#_Toc110521684)

[Gambar 2. 3 Hasil Testing pada Create data Obat 12](#_Toc110521685)

[Gambar 2. 4 Hasil Testing pada Delete data Obat 13](#_Toc110521686)

[Gambar 2. 5 Hasil Testing pada GET Golongan Obat 14](#_Toc110521687)

[Gambar 2. 6 Hasil Testing pada Update Golongan Obat 14](#_Toc110521688)

[Gambar 2. 7 Hasil Testing pada Create Golongan Obat 15](#_Toc110521689)

[Gambar 2. 8 Hasil Testing pada Delete Golongan Obat 15](#_Toc110521690)

[Gambar 2. 9 Hasil Testing pada GET Karyawan 16](#_Toc110521691)

[Gambar 2. 10 Hasil Testing pada Update Karyawan 17](#_Toc110521692)

[Gambar 2. 11 Hasil Testing pada Create Karyawan 17](#_Toc110521693)

[Gambar 2. 12 Hasil Testing pada Delete Karyawan 18](#_Toc110521694)

[Gambar 2. 13 Hasil Testing pada GET Supplier 18](#_Toc110521695)

[Gambar 2. 14 Hasil Testing pada Update Supplier 19](#_Toc110521696)

[Gambar 2. 15 Hasil Testing pada Create Supplier 20](#_Toc110521697)

[Gambar 2. 16 Hasil Testing pada Delete Supplier 20](#_Toc110521698)

[Gambar 2. 17 Hasil Testing pada GET Pelanggan 21](#_Toc110521699)

[Gambar 2. 18 Hasil Testing pada Update Pelanggan 22](#_Toc110521700)

[Gambar 2. 19 Hasil Testing pada Create Pelanggan 22](#_Toc110521701)

[Gambar 2. 20 Hasil Testing pada Delete Supplier 23](#_Toc110521702)

[Gambar 2. 21 Hasil Testing pada GET Transaksi Penjualan 23](#_Toc110521703)

[Gambar 2. 22 Hasil Testing pada Update Transaksi Penjualan 24](#_Toc110521704)

[Gambar 2. 23 Hasil Testing pada Create 24](#_Toc110521705)

[Gambar 2. 24 Hasil Testing pada Delete Transaksi Penjualan 25](#_Toc110521706)

[Gambar 2. 25 Hasil Testing pada GET Transaksi Penjualan 25](#_Toc110521707)

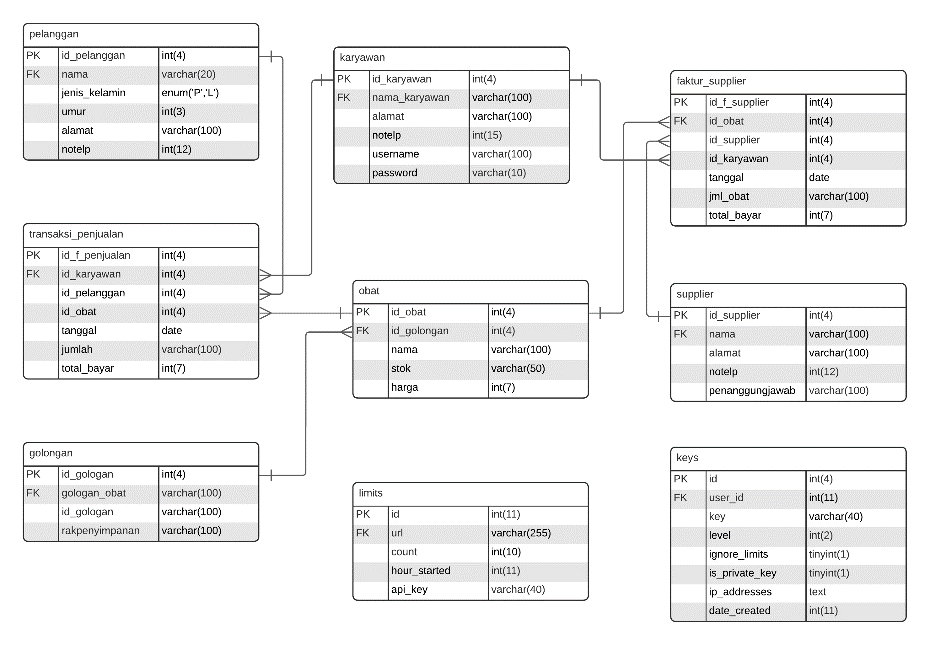
[Gambar 2. 26 Hasil Testing pada Update Transaksi Supplier 26](#_Toc110521708)

[Gambar 2. 27 Hasil Testing pada Create 27](#_Toc110521709)

[Gambar 2. 28 Hasil Testing pada Delete Transaksi Penjualan 27](#_Toc110521710)

# Diagram ERD

ERD (Entity Relationship Diagram) biasa disebut dengan diagram relasi antar entitas atau diagram hubungan entitas, dimana diagram ERD digunakan untuk perancangan suatu database dan menunjukan relasi antar objek atau entitas beserta atribut-atributnya secara detail. Beriut ini merupakan diagram ERD mengenai Apotek:



Gambar 1. 1 Diagram ERD

Pada gambar diagram ERD diatas, dapat dilihat terdapat 9 (Sembilan) tabel yang saling berelasi. Berikut keterangan setiap tabel pada diagram ERD:

## Tabel Golongan

Tabel Golongan merupakan tabel yang digunakan untuk mengklasifikasikan obat berdasarkan kegunaan obat. Tabel golongan juga digunakan untuk mengetahui dimana tempat penyimpanan obat. Tabel golongan terdiri 4 field dengan 1 Primary Key yaitu id\_golongan. Berikut Field field dengan type data yang dimiliki Tabel Golongan:

Tabel 1. 1 Tabel Golongan

|  |  |
| --- | --- |
| **Field** | **Type** |
| golongan\_obat | varchar (100) |
| kegunaan | varchar (100) |
| id\_golongan | int (4) |
| rakpenyimpanan | varchar (100) |

## Tabel Obat

Tabel Obat merupakan tabel yang digunakan untuk mendata semua obat yang terdapat di Apotek. Pendataan obat dilakukan untuk mengetahui ketersediaan stok serta harga jual obat. Tabel obat terdiri dari 5 field dengan 1 Primary Key yaitu id\_obat dan 1 Foreign Key yaitu id\_golongan. Berikut Field field dengan type data yang dimiliki Tabel obat:

Tabel 1. 2 Tabel Obat

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_obat | int (4) |
| id\_golongan | int (4) |
| nama\_obat | varchar (100) |
| stok | int (50) |
| harga | int (7) |

## Tabel Transaksi Penjualan

Tabel transaksi\_penjualan merupakan tabel yang mendata penjualan obat kepada pelanggan. Dengan tabel transaksi\_penjualan dapat diketahui obat apa yang dibeli pelanggan, berapa jumlah pembelian obat, serta total pembayaran yang harus dibayar pelanggan. Tabel transaksi\_penjualan terdiri dari 7 field dengan 1 Primary Key yaitu id\_t\_penjualan dan 3 Foreign Key yaitu id\_karyawan, id\_pelanggan, dan id\_obat. Berikut Field field dengan type data yang dimiliki Tabel transaksi\_penjualan:

Tabel 1. 3 Tabel Transaksi Penjualan

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_t\_penjualan | int (4) |
| id\_karyawan | int (4) |
| id\_pelanggan | int (4) |
| id\_obat | int (4) |
| tanggal | date |
| jumlah | int (100) |
| total\_bayar | int (7) |

## Tabel Transaksi Supplier

Tabel transaksi\_supplier merupakan tabel yang mendata pemesanan obat kepada supplier. Dengan tabel transaksi\_supplier dapat diketahui obat apa yang dipesan, berapa jumlah pemesanan obat, serta total pembayaran yang harus dibayar dalam pemesanan obat. Tabel transaksi\_supplier terdiri dari 7 field dengan 1 Primary Key yaitu id\_t\_supplier dan 3 Foreign Key yaitu id\_obat, id\_supplier, dan id\_karyawan. Berikut Field field dengan type data yang dimiliki Tabel transaksi\_supplier:

Tabel 1. 4 Tabel Transaksi Supplier

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_t\_supplier | int (4) |
| id\_obat | int (4) |
| id\_supplier | int (4) |
| id\_karyawan | int (4) |
| tanggal | date |
| jumlah | int (100) |
| total\_bayar | int (7) |

## Tabel Karyawan

Tabel karyawan merupakan tabel yang digunakan untuk mendata karyawan yang bekerja di Apotek. Pendataan karyawan dilakukan untuk mengetahui alamat serta nomor telepon karyawan. Tabel karyawan terdiri dari 6 field dengan 1 Primary Key yaitu id\_karyawan. Berikut Field field dengan type data yang dimiliki Tabel karyawan:

Tabel 1. 5 Tabel Karyawan

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_karyawan | int (4) |
| nama\_karyawan | varchar (100) |
| alamat | varchar (100) |
| notelp | int (15) |
| user | varchar (100) |
| password | varchar (10) |

## Tabel Pelanggan

Tabel pelanggan merupakan tabel yang digunakan untuk mendata pelanggan yang yang membeli obat di Apotek. Pendataan pelanggan dilakukan untuk mengetahui umur, alamat serta nomor telepon pelanggan. Tabel pelanggan terdiri dari 6 field dengan 1 Primary Key yaitu id\_pelanggan. Berikut Field field dengan type data yang dimiliki Tabel pelanggan:

Tabel 1. 6 Tabel Pelanggan

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_pelanggan | int (4) |
| nama\_pelanggan | varchar (100) |
| jenis\_kelamin | enum('perempuan', 'laki-laki') |
| umur | int (3) |
| alamat | varchar (100) |
| notelp | int (12) |

## Tabel Supplier

Tabel supplier merupakan tabel yang digunakan untuk mendata supplier yang bekerja sama dengan Apotek. Tabel supplier terdiri dari 5 field dengan 1 Primary Key yaitu id\_supplier. Berikut Field field dengan type data yang dimiliki Tabel supplier:

Tabel 1. 7 Tabel Supplier

|  |  |
| --- | --- |
| **Field** | **Type** |
| id\_supplier | int (4) |
| nama\_supplier | varchar (100) |
| alamat | varchar (100) |
| notelp | int (12) |
| penanggungjawab | int (12) |

## Tabel Keys

Tabel keys merupakan tabel untuk menyimpan API KEY, dimana API KEY berfungsi untuk memberikan akses login dan mencegah tindakan penyalahgunaan API. Tabel keys terdiri dari 8 field dengan 1 Primary Key yaitu id. Berikut Field field dengan type data yang dimiliki Tabel supplier:

Tabel 1. 8 Tabel Keys

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int (11) |
| user\_id | int (11) |
| key | varchar (40) |
| level | int (2) |
| ignore\_limits | tinyint (1) |
| is\_private\_key | tinyint (1) |
| ip\_addresses | text |
| date\_created | int (11) |

## Tabel Limits

Tabel limits digunakan untuk pembatasan data baik pada saat pencarian data atau penginputan data. Tabel keys terdiri dari 5 field dengan 1 Primary Key yaitu id. Berikut Field field dengan type data yang dimiliki Tabel limits:

Tabel 1. 9 Tabel Limits

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int (11) |
| uri | varchar (255) |
| count | int (0) |
| **hour\_started** | int (11) |
| **api\_key** | varchar (40) |

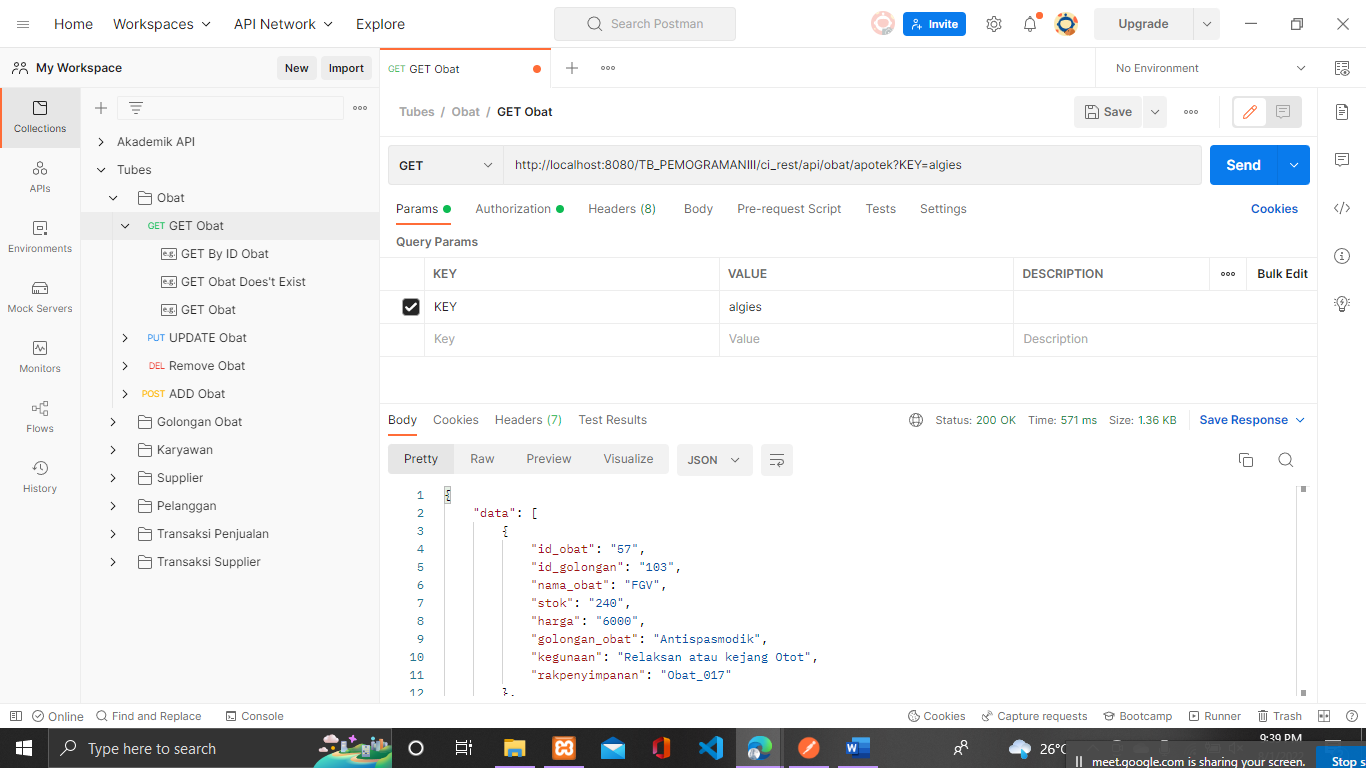
# Testing Masing-Masing Endpoint Pada Rest Server Menggunakan POSTMAN

## Endpoint dan Hasil Testing pada Rest Server Obat

### **Hasil Testing pada GET Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data obat:

http://localhost:8080/TB\_PEMOGRAMANIII /ci\_rest/api/obat/apotek?KEY=algies

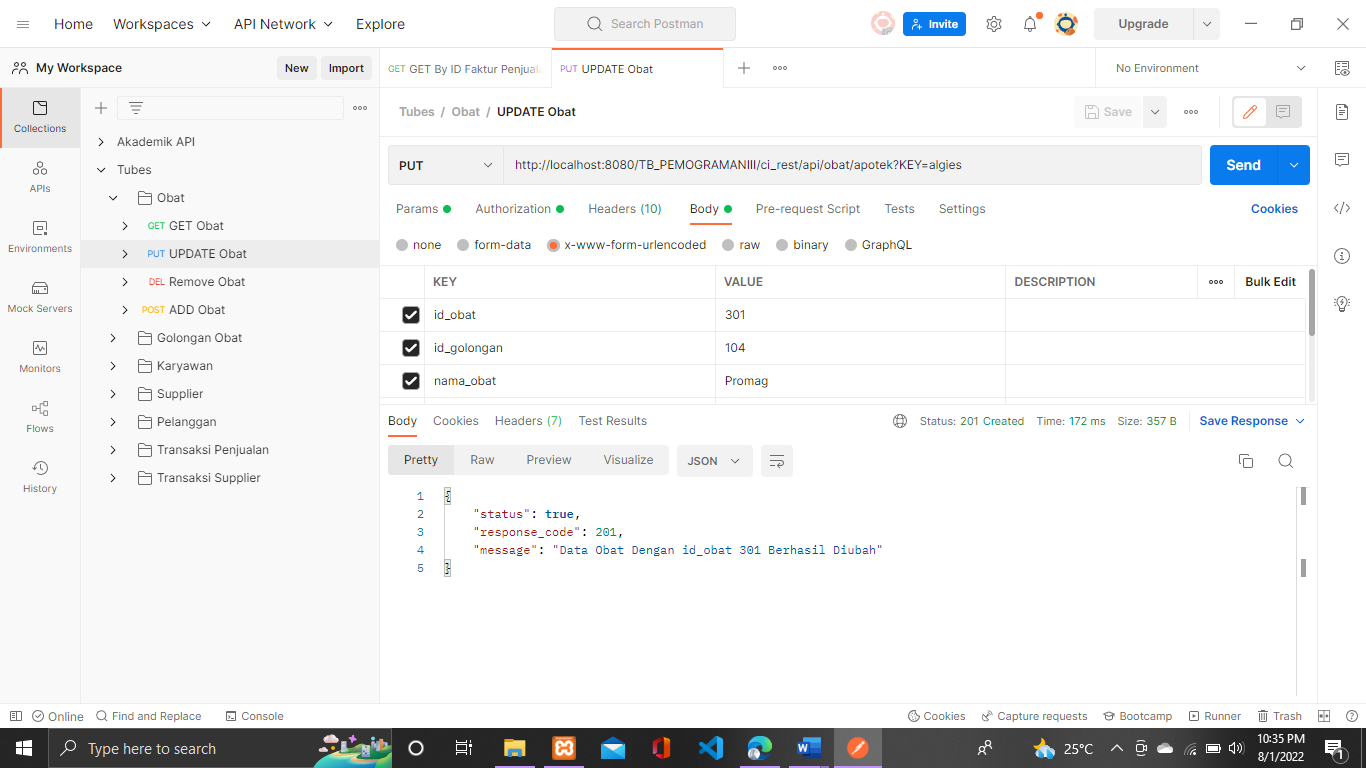


Gambar 2. 1 Hasil Testing pada GET Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint obat yang diuji untuk menampilkan seluruh data obat. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint obat metode yang digunakan untuk memanggil data obat adalah dengan metode GET. Dalam menampilkan data obat, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update data Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Obat: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/obat/apotek?KEY=algies

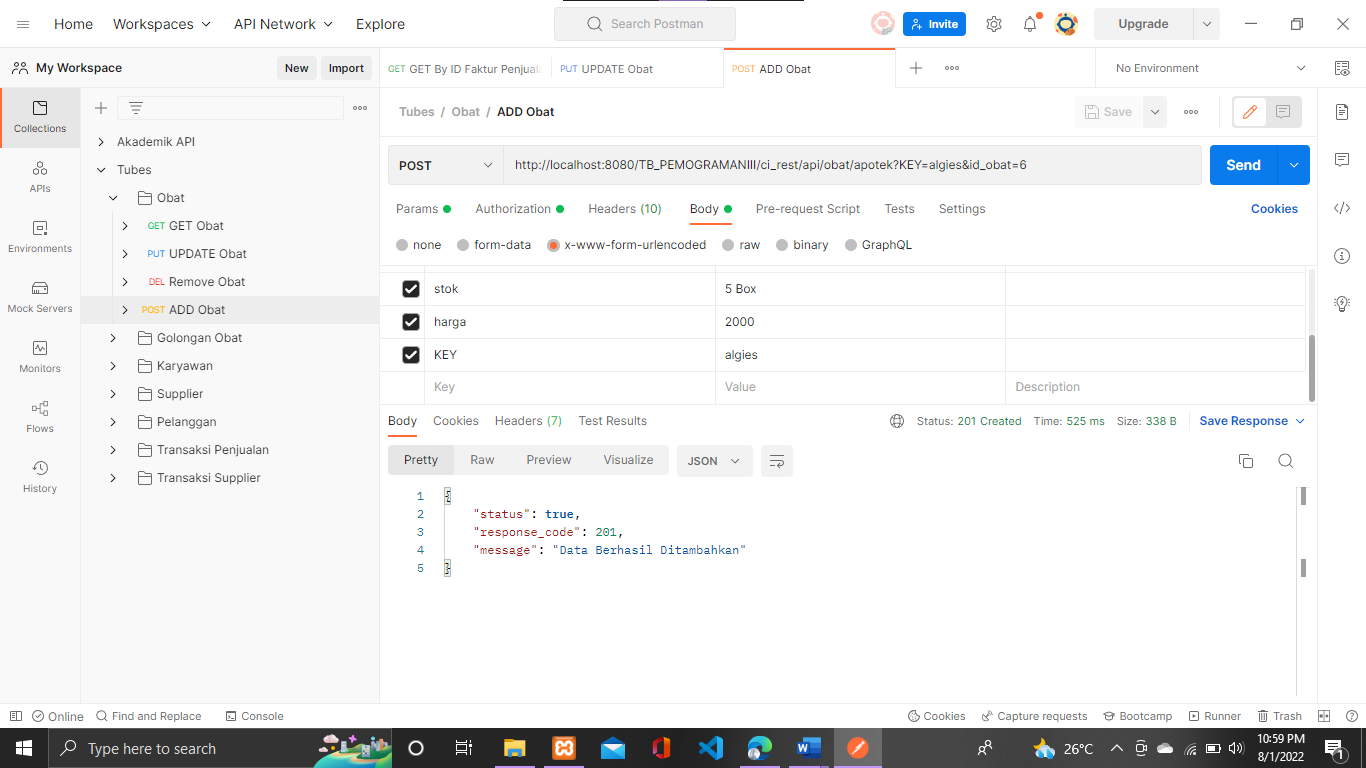


Gambar 2. 2 Hasil Testing pada Update data Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint obat yang diuji untuk mengedit data obat. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint obat, metode yang digunakan untuk mengedit data obat adalah dengan metode PUT.

### **Hasil Testing pada Create data Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Obat: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/obat/apotek?KEY=algies&id\_obat=6

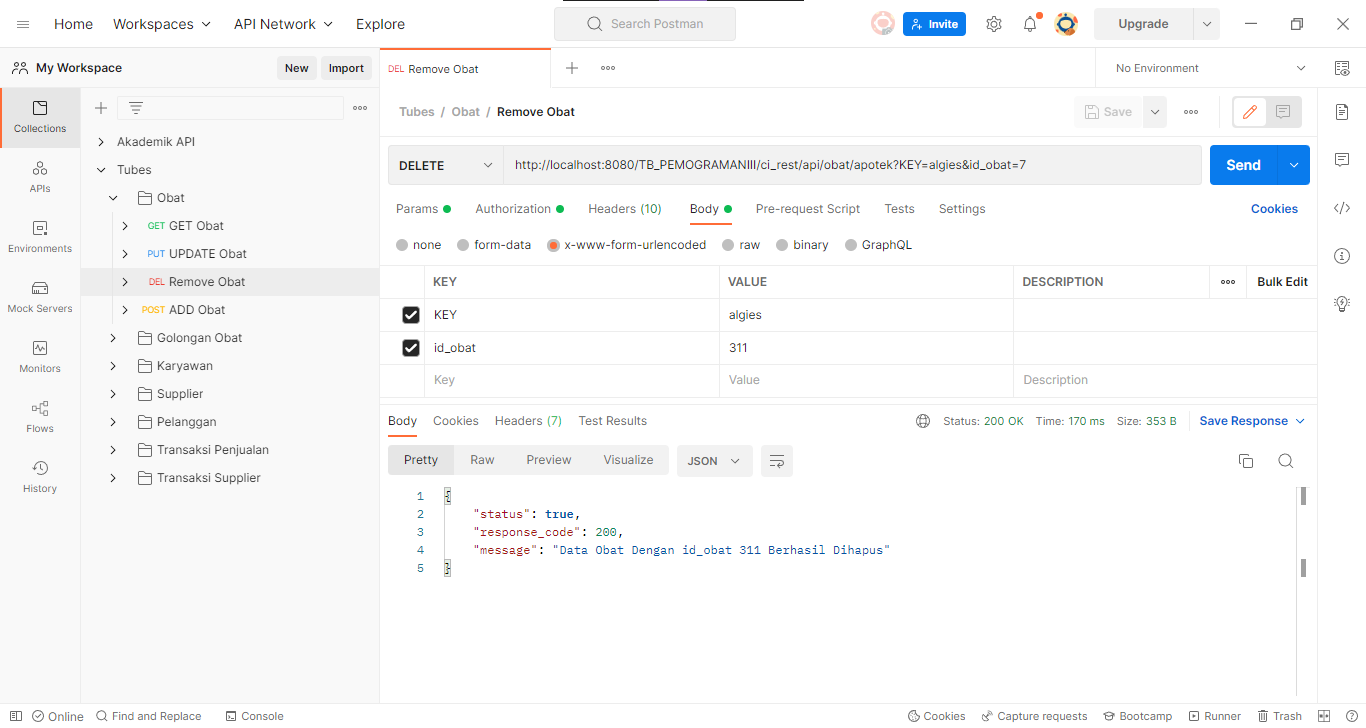


Gambar 2. 3 Hasil Testing pada Create data Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint obat yang diuji untuk menambah data obat. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint obat, metode yang digunakan untuk menambah data obat adalah dengan metode POST.

### **Hasil Testing pada Delete data Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Obat: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/obat/apotek?KEY=algies&id\_obat=311



Gambar 2. 4 Hasil Testing pada Delete data Obat

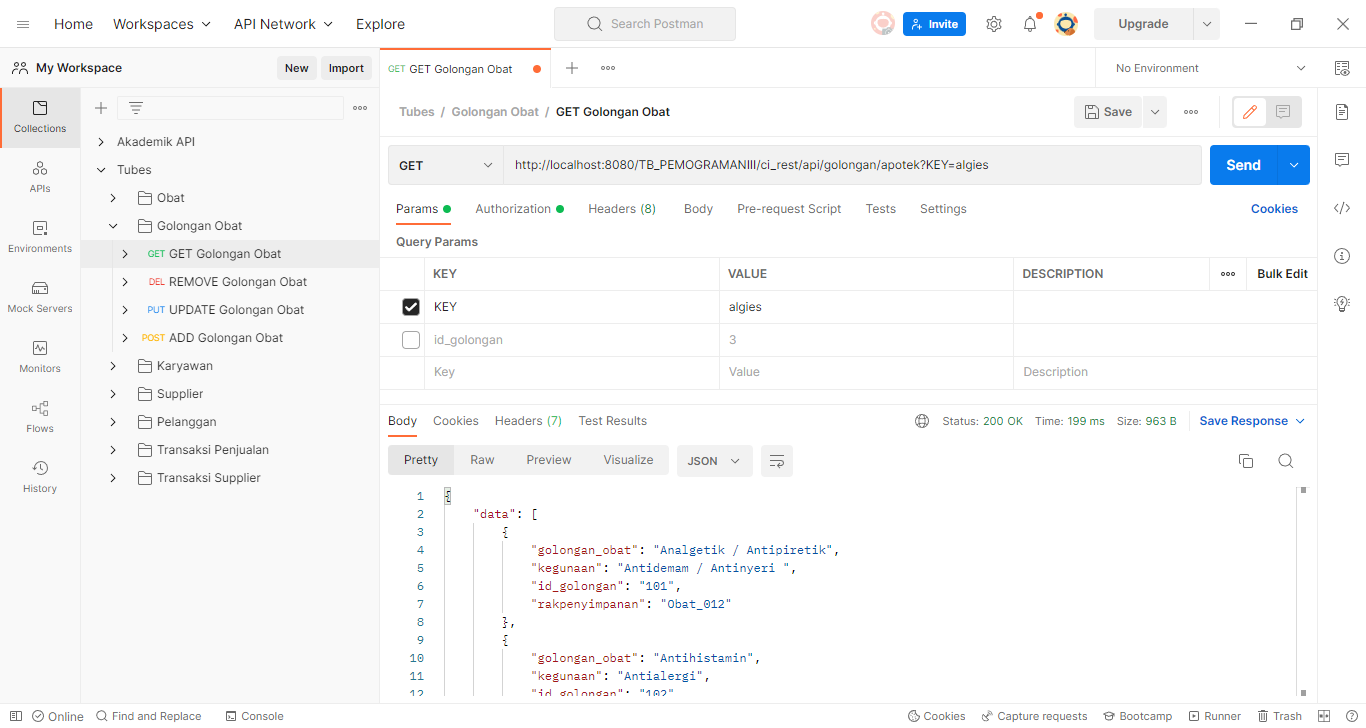
Pada gambar diatas, dapat dilihat mengenai Endpoint obat yang diuji untuk menghapus data obat. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint obat, metode yang digunakan untuk menghapus data obat adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Golongan Obat

### **Hasil Testing pada GET Golongan Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data golongan obat:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/golongan/apotek?KEY=algies



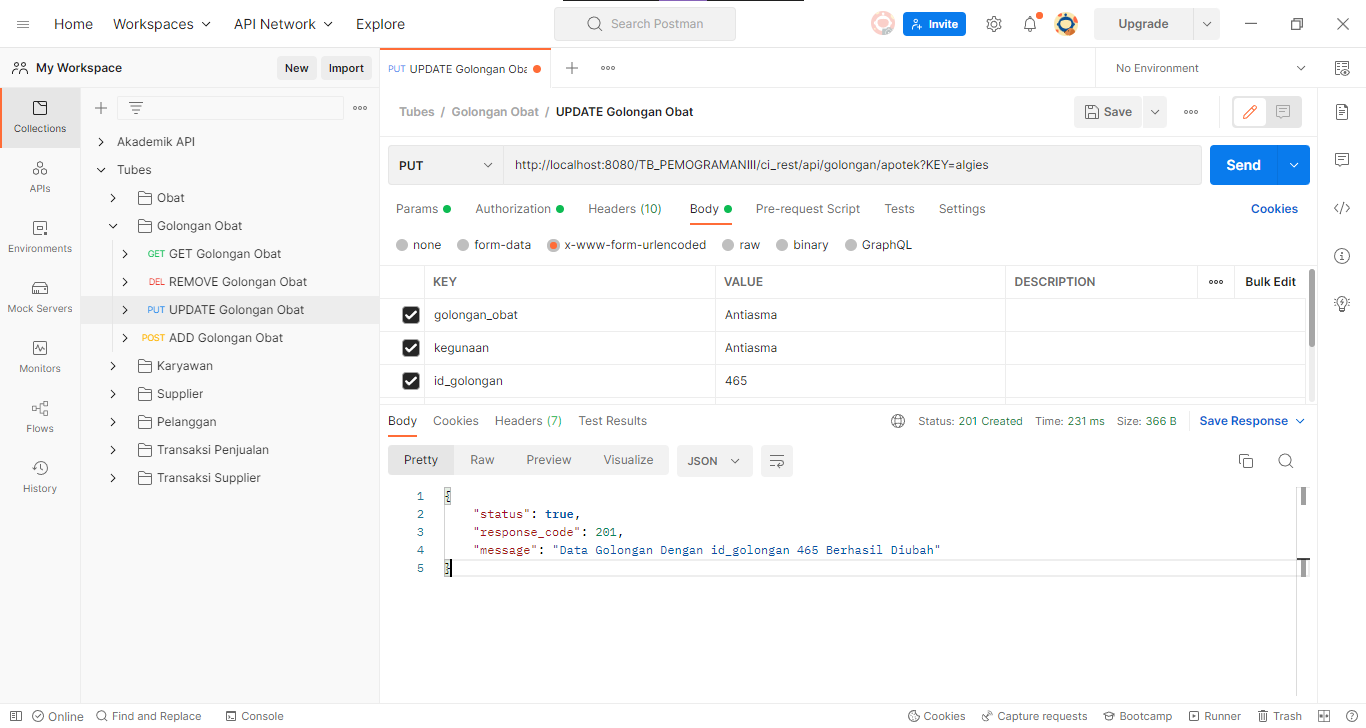
Gambar 2. 5 Hasil Testing pada GET Golongan Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint golongan obat yang diuji untuk menampilkan seluruh data golongan obat. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint golongan obat metode yang digunakan untuk memanggil data golongan obat adalah dengan metode GET. Dalam menampilkan data golongan obat, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Golongan Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Golongan Obat:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/golongan/apotek?KEY=algies



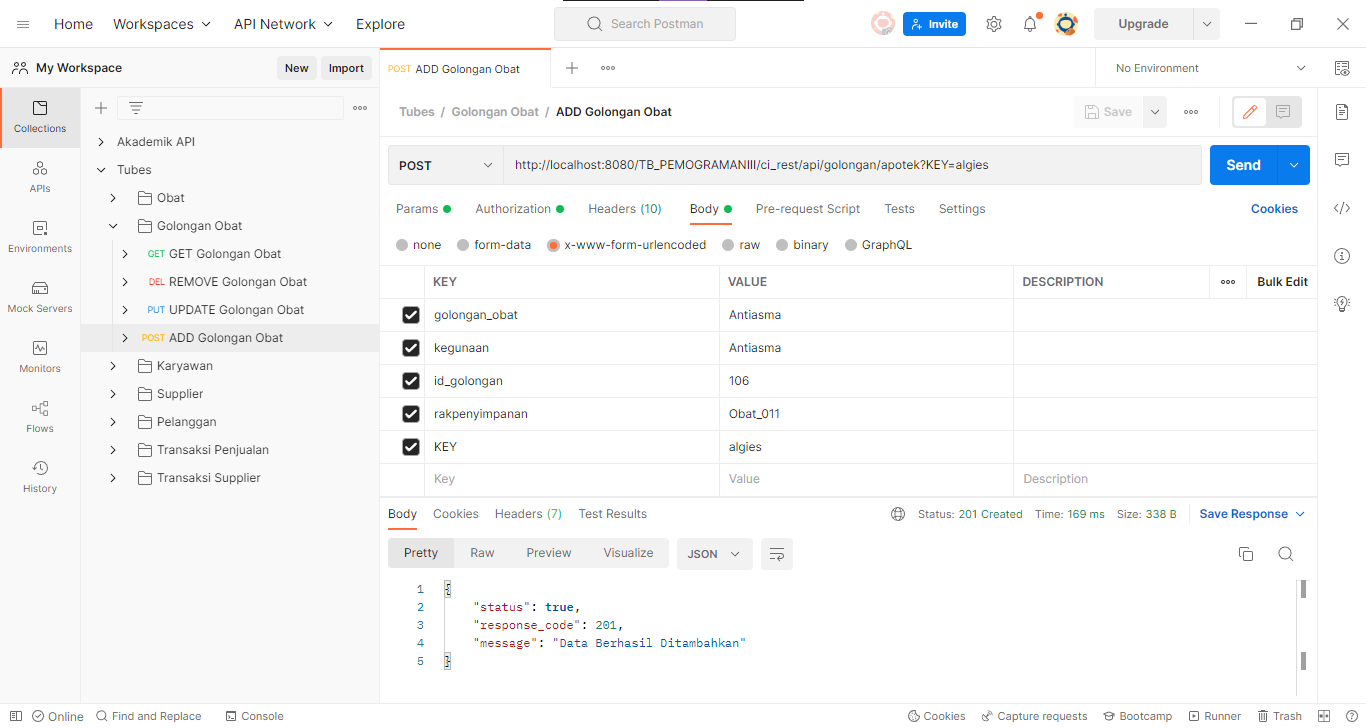
Gambar 2. 6 Hasil Testing pada Update Golongan Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint Golongan obat yang diuji untuk mengedit data Golongan obat. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Golongan obat, metode yang digunakan untuk mengedit data Golongan obat adalah dengan metode PUT.

### **Hasil Testing pada Create Golongan Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Golongan Obat:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/golongan/apotek?KEY=algies



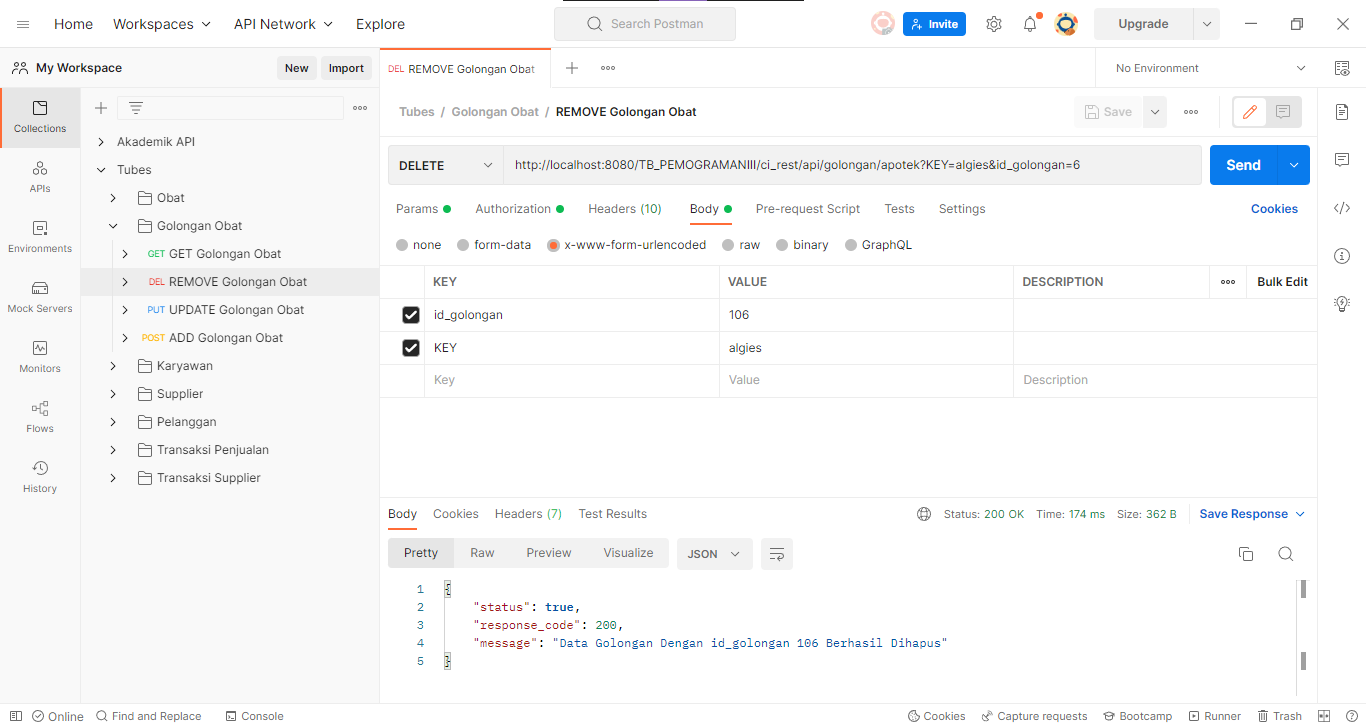
Gambar 2. 7 Hasil Testing pada Create Golongan Obat

Pada gambar diatas, dapat dilihat mengenai Endpoint Golongan obat yang diuji untuk menambah data Golongan obat. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Golongan obat, metode yang digunakan untuk menambah data Golongan obat adalah dengan metode POST.

### **Hasil Testing pada Delete Golongan Obat**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data golongan Obat:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/golongan/apotek?KEY=algies&id\_golongan=106



Gambar 2. 8 Hasil Testing pada Delete Golongan Obat

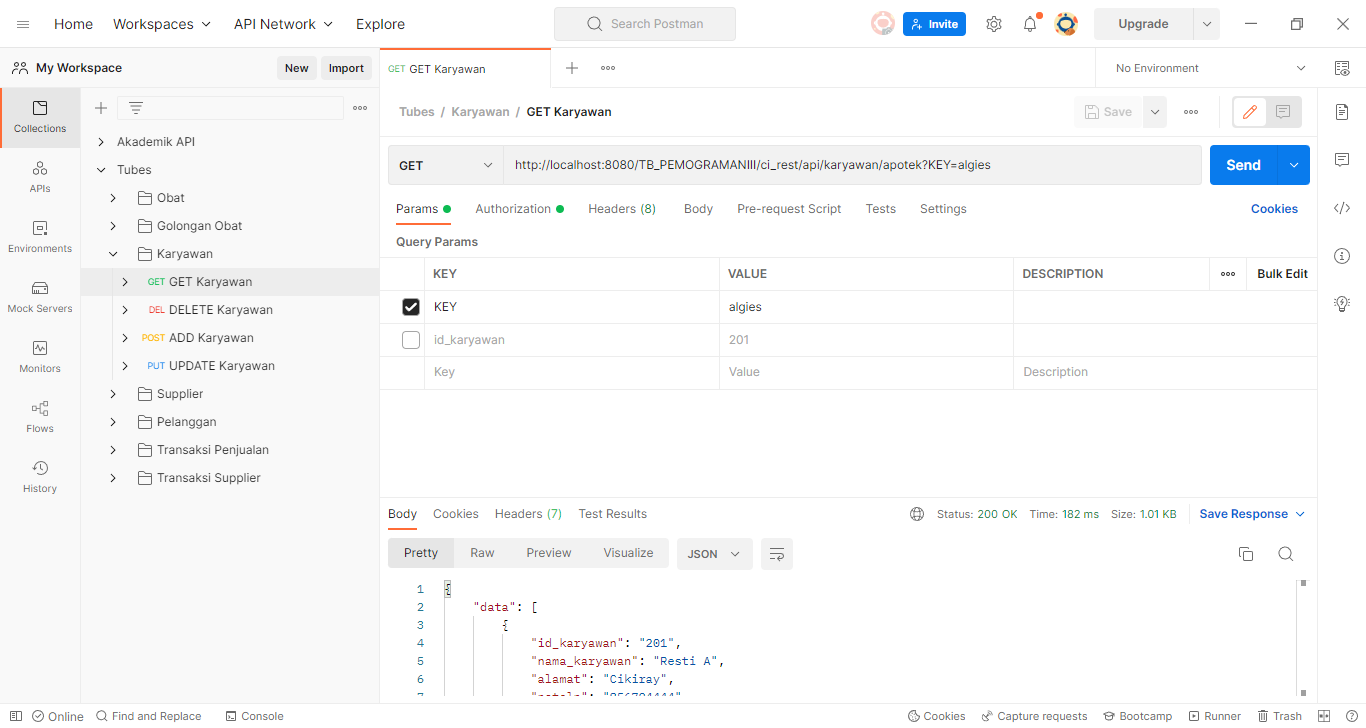
Pada gambar diatas, dapat dilihat mengenai Endpoint golongan obat yang diuji untuk menghapus data golongan obat. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint golongan obat, metode yang digunakan untuk menghapus data golongan obat adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Karyawan

### **Hasil Testing pada GET Karyawan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data Karyawan:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/karyawan/apotek?KEY=algies



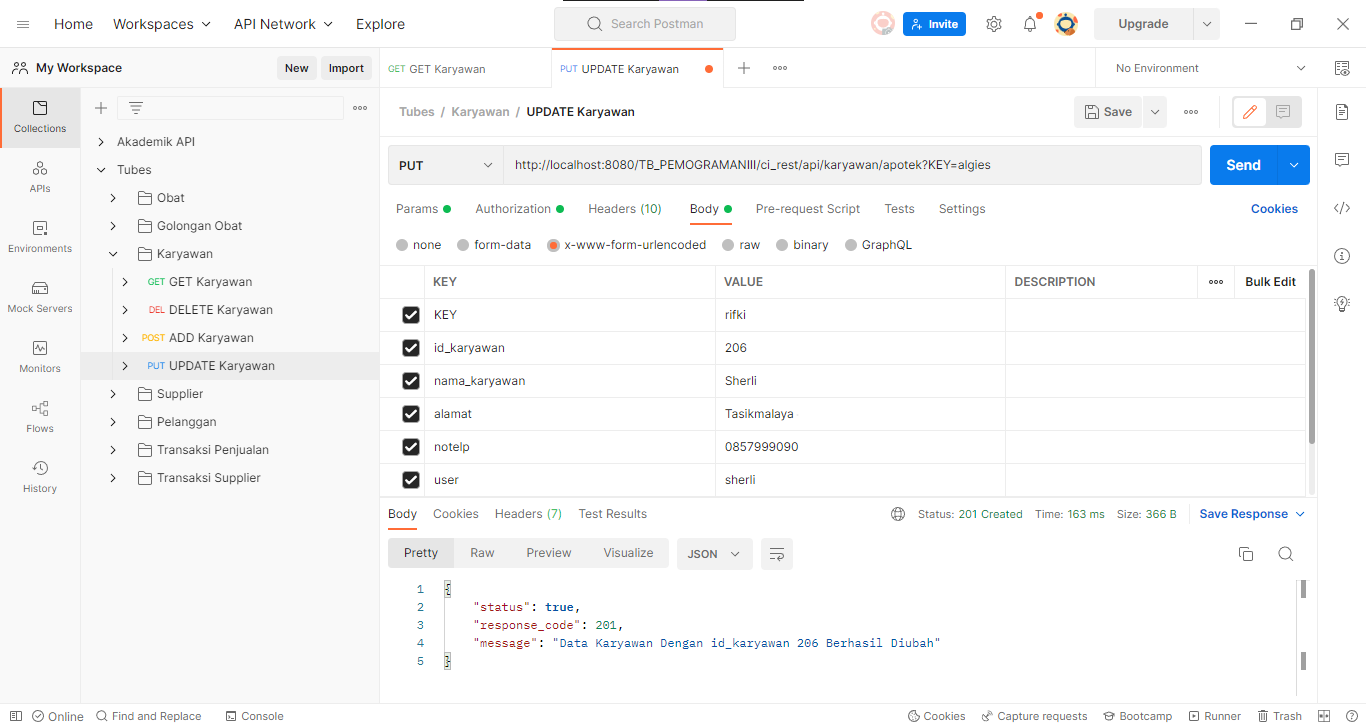
Gambar 2. 9 Hasil Testing pada GET Karyawan

Pada gambar diatas, dapat dilihat mengenai Endpoint Karyawan yang diuji untuk menampilkan seluruh data Karyawan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Karyawan metode yang digunakan untuk memanggil data Karyawan adalah dengan metode GET. Dalam menampilkan data Karyawan, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Karyawan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Karyawan:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/karyawan/apotek?KEY=algies



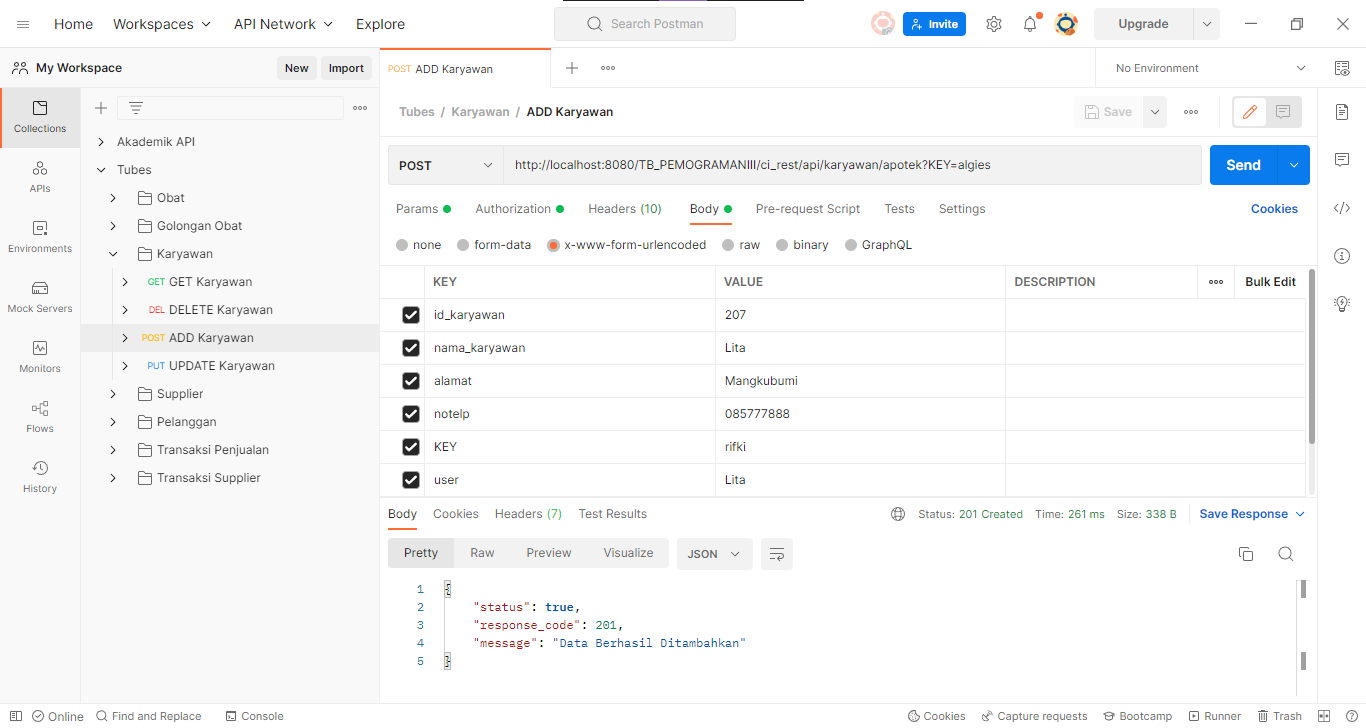
Gambar 2. 10 Hasil Testing pada Update Karyawan

Pada gambar diatas, dapat dilihat mengenai Endpoint Karyawan yang diuji untuk mengedit data Karyawan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Karyawan, metode yang digunakan untuk mengedit data Karyawan adalah dengan metode PUT.

### **Hasil Testing pada Create Karyawan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Karyawan:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/karyawan/apotek?KEY=algies

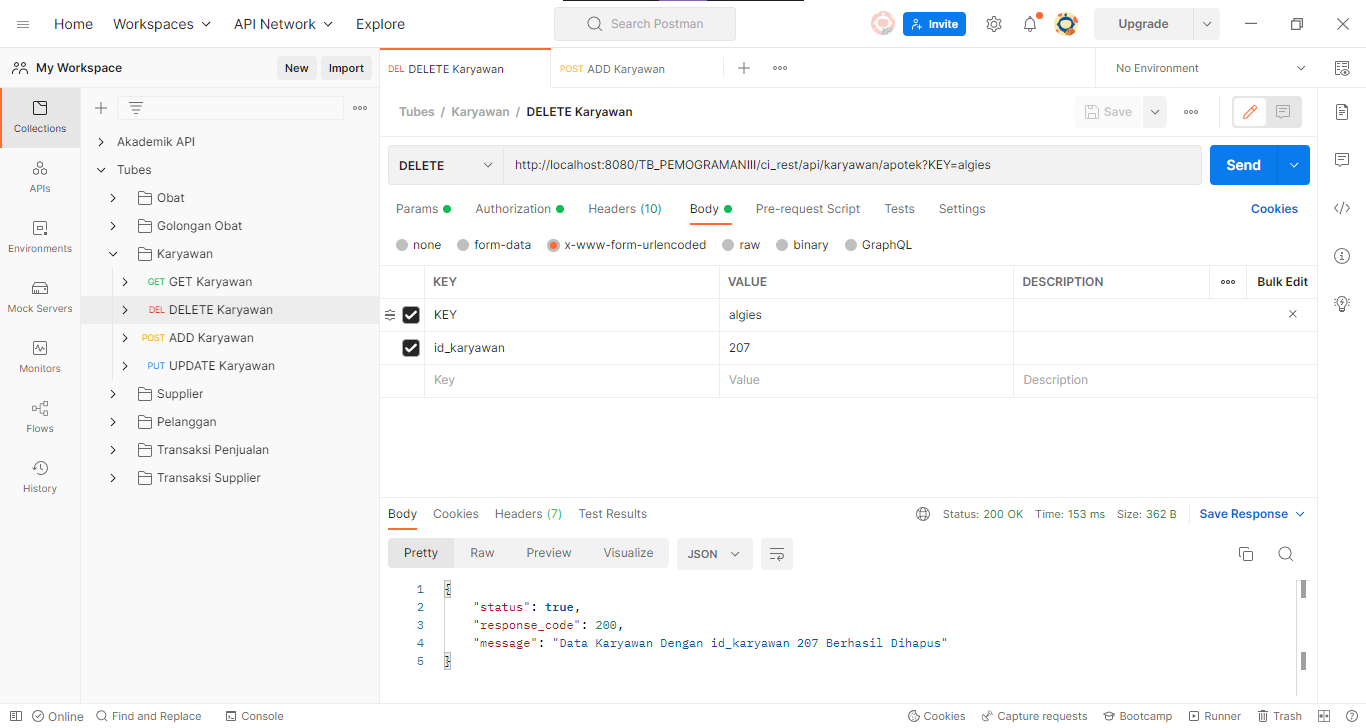


Gambar 2. 11 Hasil Testing pada Create Karyawan

Pada gambar diatas, dapat dilihat mengenai Endpoint Karyawan yang diuji untuk menambah data Karyawan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Karyawan, metode yang digunakan untuk menambah data Karyawan adalah dengan metode POST.

### **Hasil Testing pada Delete Karyawan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Karyawan: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/karyawan/apotek?KEY=algies



Gambar 2. 12 Hasil Testing pada Delete Karyawan

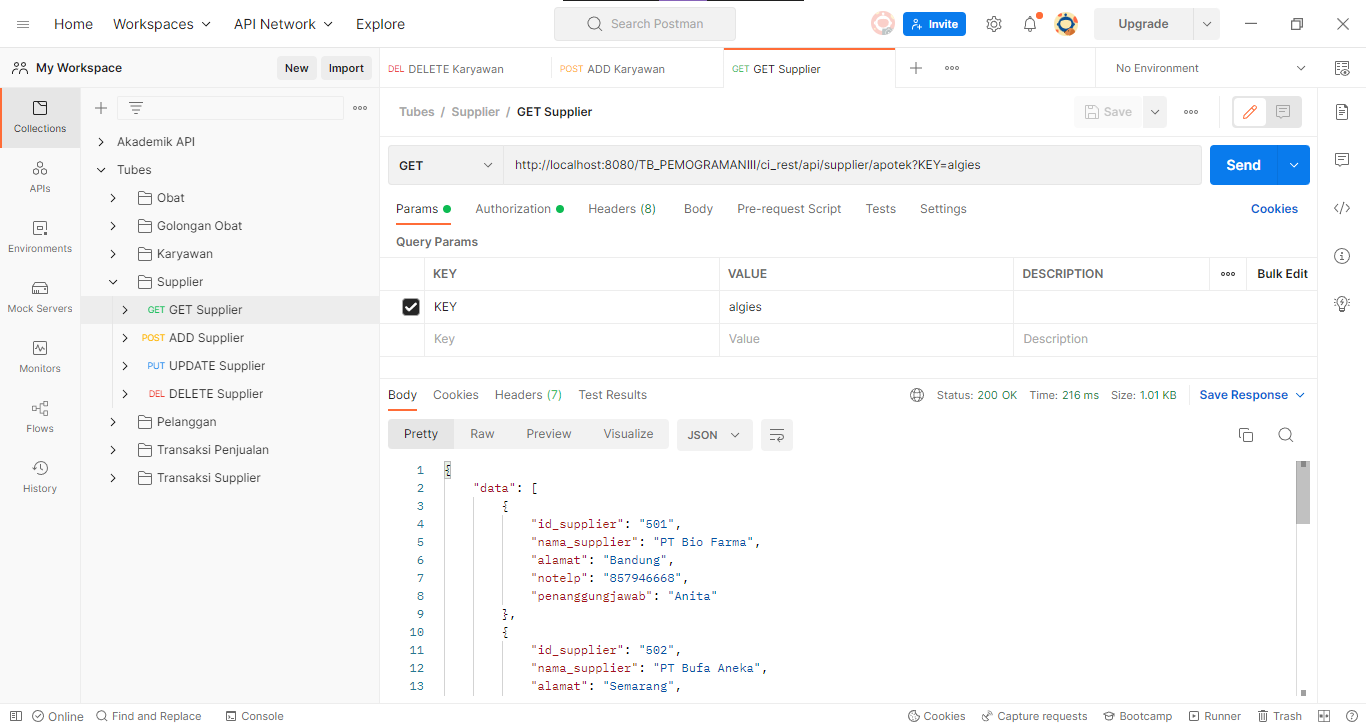
Pada gambar diatas, dapat dilihat mengenai Endpoint Karyawan yang diuji untuk menghapus data Karyawan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Karyawan, metode yang digunakan untuk menghapus data Karyawan adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Supplier

### **Hasil Testing pada GET Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data Supplier:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/supplier/apotek?KEY=algies



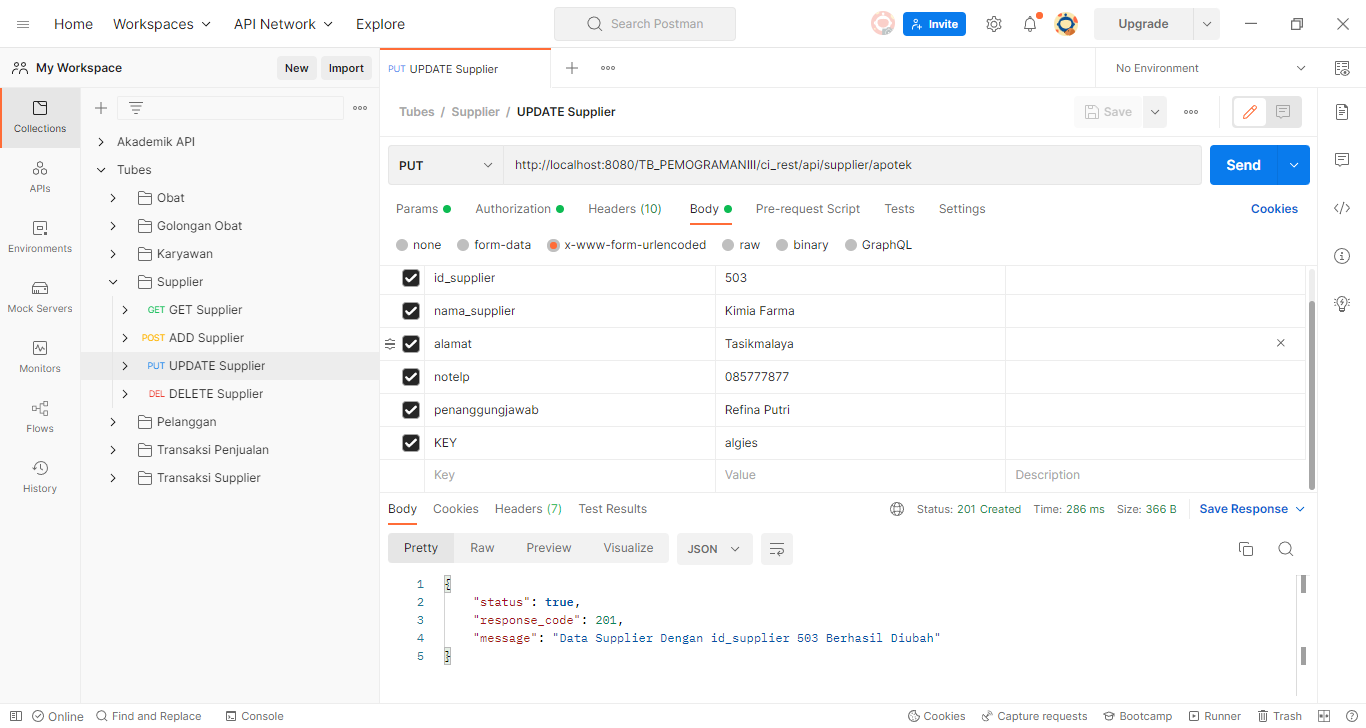
Gambar 2. 13 Hasil Testing pada GET Supplier

Pada gambar diatas, dapat dilihat mengenai Endpoint Supplier yang diuji untuk menampilkan seluruh data Supplier. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Supplier metode yang digunakan untuk memanggil data Supplier adalah dengan metode GET. Dalam menampilkan data Supplier, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Supplier:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/supplier/apotek



Gambar 2. 14 Hasil Testing pada Update Supplier

Pada gambar diatas, dapat dilihat mengenai Endpoint Supplier yang diuji untuk mengedit data Supplier. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Supplier, metode yang digunakan untuk mengedit data Supplier adalah dengan metode PUT.

### **Hasil Testing pada Create Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Supplier:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/supplier/apotek?KEY=rifki

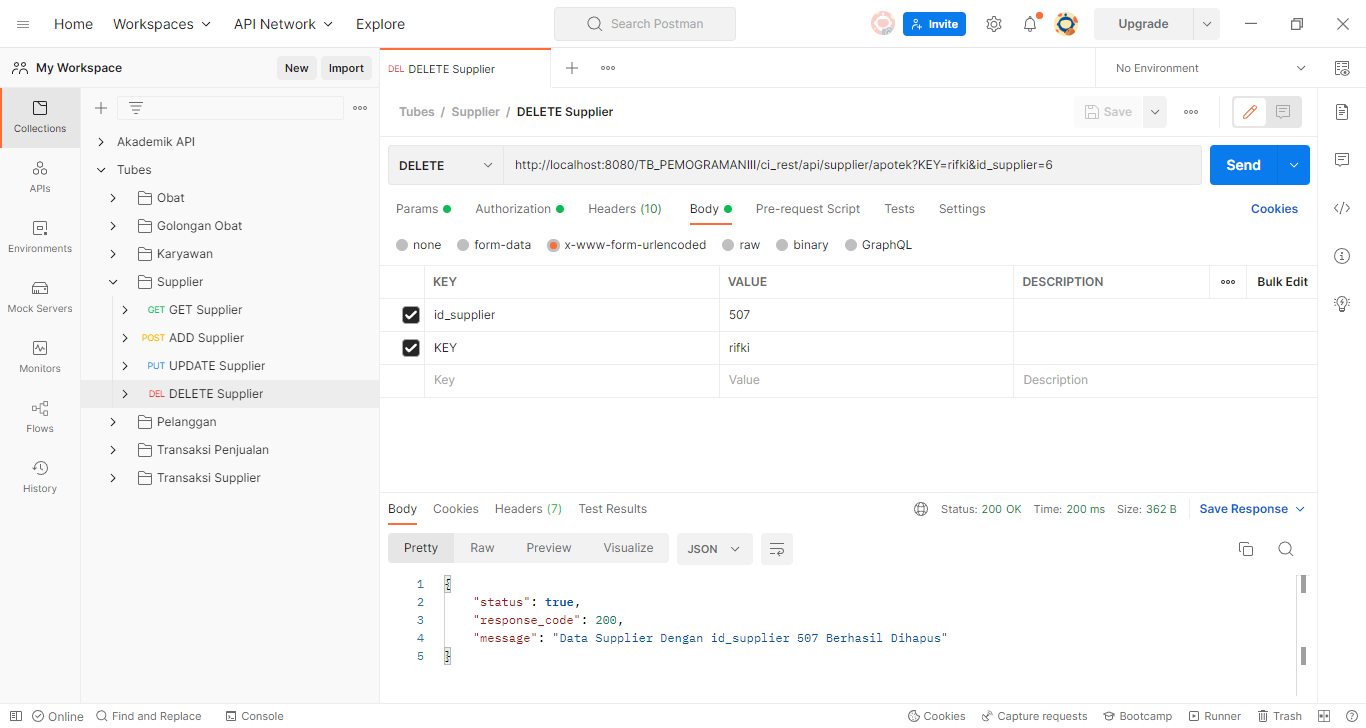


Gambar 2. 15 Hasil Testing pada Create Supplier

Pada gambar diatas, dapat dilihat mengenai Endpoint Supplier yang diuji untuk menambah data Supplier. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Supplier, metode yang digunakan untuk menambah data Supplier adalah dengan metode POST.

### **Hasil Testing pada Delete Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Supplier: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/supplier/apotek?KEY=rifki&id\_supplier=507



Gambar 2. 16 Hasil Testing pada Delete Supplier

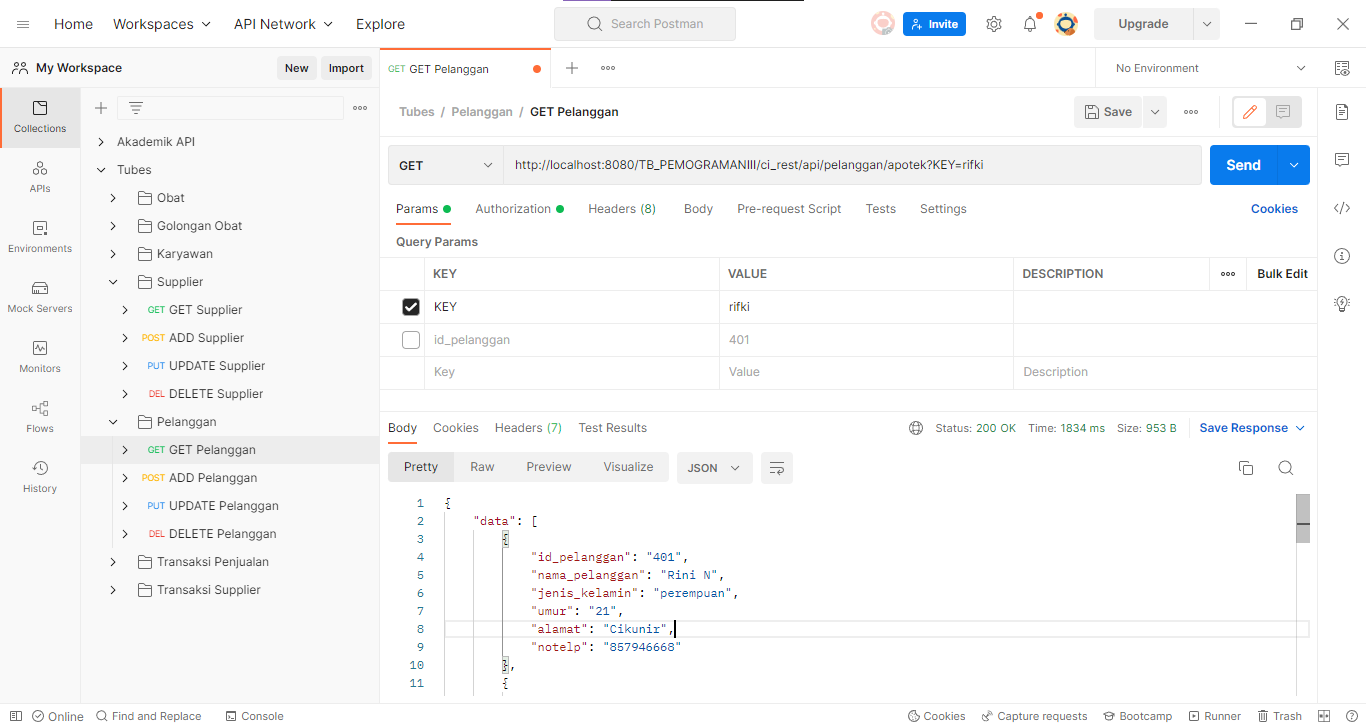
Pada gambar diatas, dapat dilihat mengenai Endpoint Supplier yang diuji untuk menghapus data Supplier. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Supplier, metode yang digunakan untuk menghapus data Supplier adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Pelanggan

### **Hasil Testing pada GET Pelanggan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data Pelanggan:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/pelanggan/apotek?KEY=rifki>

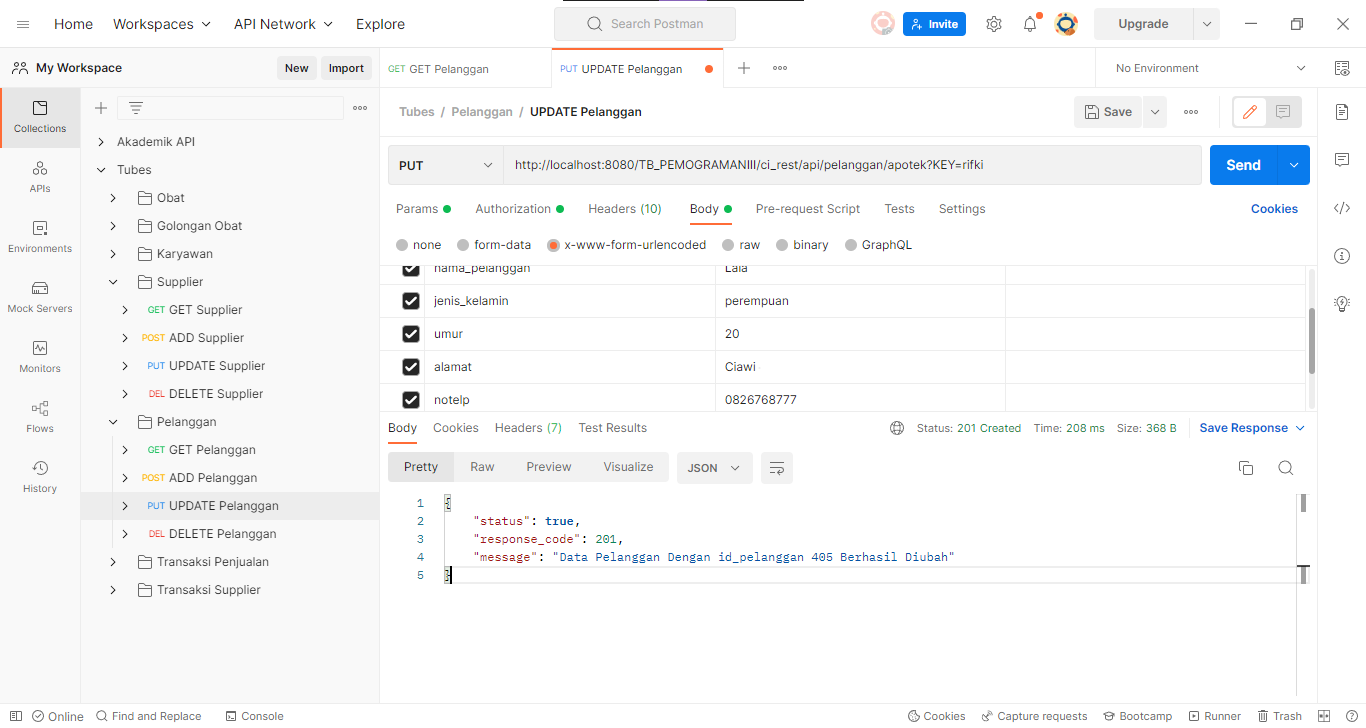


Gambar 2. 17 Hasil Testing pada GET Pelanggan

Pada gambar diatas, dapat dilihat mengenai Endpoint Pelanggan yang diuji untuk menampilkan seluruh data Pelanggan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Pelanggan metode yang digunakan untuk memanggil data Pelanggan adalah dengan metode GET. Dalam menampilkan data Pelanggan, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Pelanggan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Pelanggan: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/pelanggan/apotek?KEY=rifki



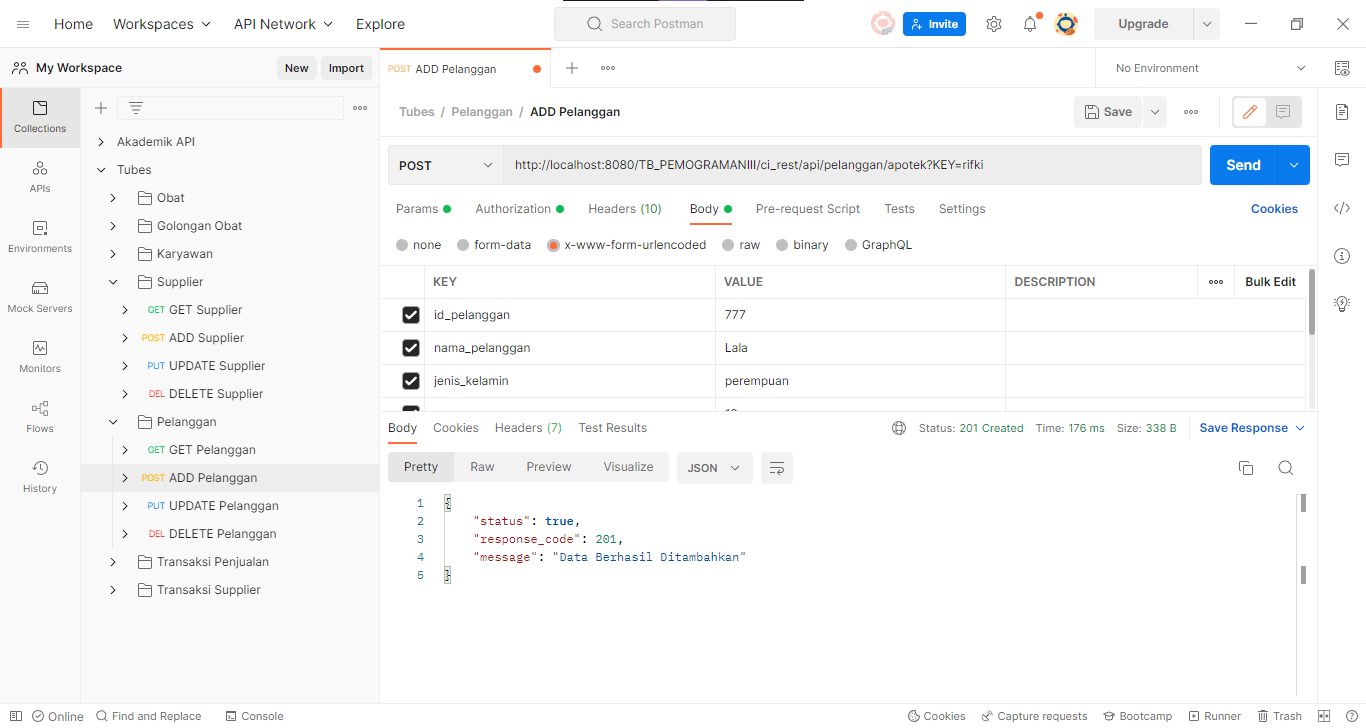
Gambar 2. 18 Hasil Testing pada Update Pelanggan

Pada gambar diatas, dapat dilihat mengenai Endpoint Pelanggan yang diuji untuk mengedit data Pelanggan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Supplier, metode yang digunakan untuk mengedit data Pelanggan adalah dengan metode PUT.

### **Hasil Testing pada Create Pelanggan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Pelanggan:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/pelanggan/apotek?KEY=rifki>



Gambar 2. 19 Hasil Testing pada Create Pelanggan

Pada gambar diatas, dapat dilihat mengenai Endpoint Pelanggan yang diuji untuk menambah data Pelanggan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Pelanggan, metode yang digunakan untuk menambah data Supplier adalah dengan metode POST.

### **Hasil Testing pada Delete Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Pelanggan: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/supplier/apotek?KEY=rifki&id\_supplier=507



Gambar 2. 20 Hasil Testing pada Delete Supplier

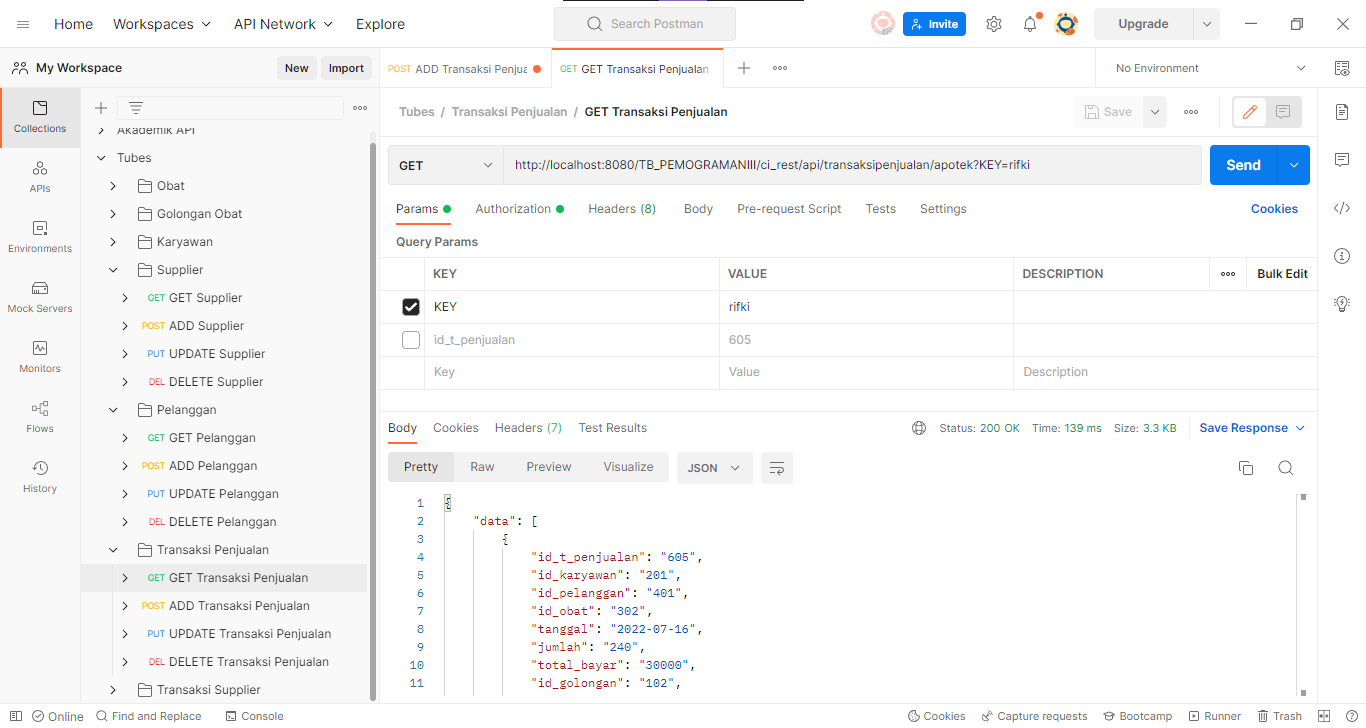
Pada gambar diatas, dapat dilihat mengenai Endpoint Pelanggan yang diuji untuk menghapus data Pelanggan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Pelanggan, metode yang digunakan untuk menghapus data Pelanggan adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Transaksi Penjualan

### **Hasil Testing pada GET Transaksi Penjualan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data Transaksi Penjualan:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/transaksipenjualan/apotek?KEY=rifki>



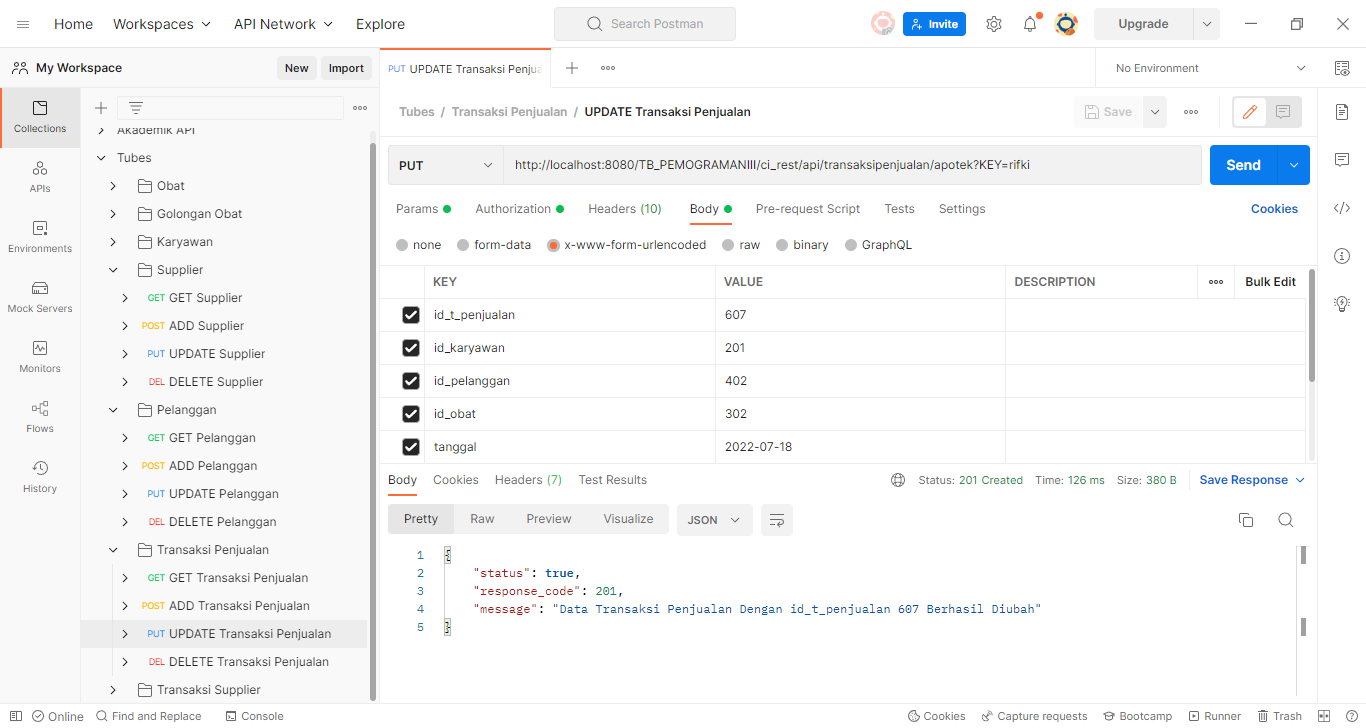
Gambar 2. 21 Hasil Testing pada GET Transaksi Penjualan

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Penjualan yang diuji untuk menampilkan seluruh data Transaksi Penjualan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Transaksi Penjualan, metode yang digunakan untuk memanggil data Pelanggan adalah dengan metode GET. Dalam menampilkan data Transaksi Penjualan, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Transaksi Penjualan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Transaksi Penjualan:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/transaksipenjualan/apotek?KEY=rifki



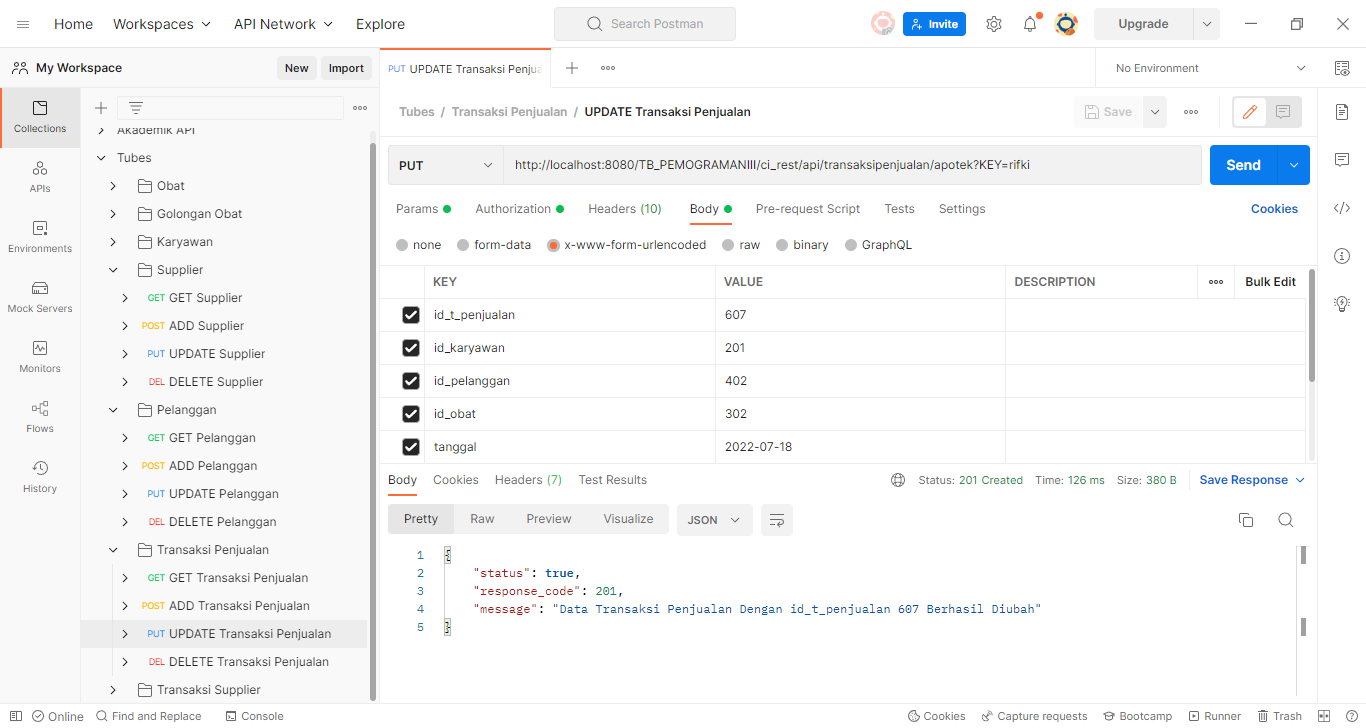
Gambar 2. 22 Hasil Testing pada Update Transaksi Penjualan

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Penjualanyang diuji untuk mengedit data Transaksi Penjualan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Supplier, metode yang digunakan untuk mengedit data Transaksi Penjualan adalah dengan metode PUT.

### **Hasil Testing pada Create**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Pelanggan:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/transaksipenjualan/apotek?KEY=rifki>

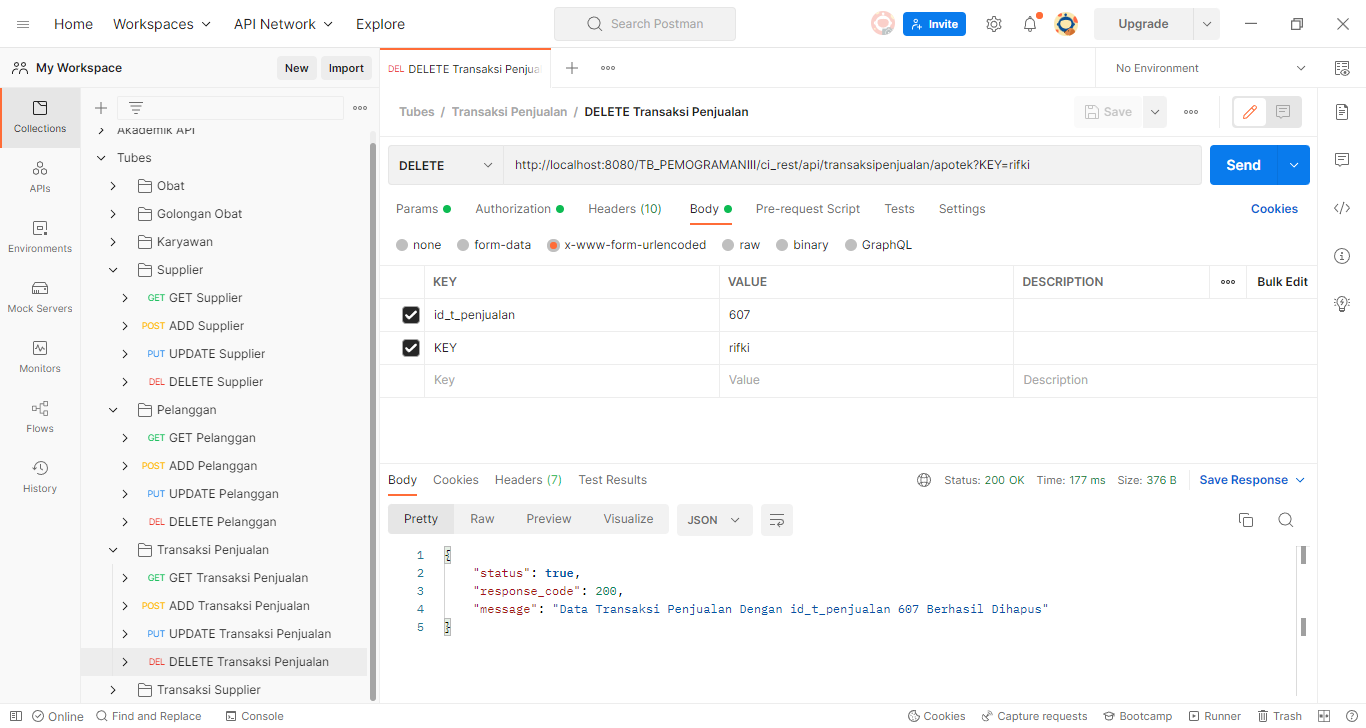
****

Gambar 2. 23 Hasil Testing pada Create

Pada gambar diatas, dapat dilihat mengenai Endpoint Pelanggan yang diuji untuk menambah data Pelanggan. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Pelanggan, metode yang digunakan untuk menambah data Supplier adalah dengan metode POST.

### **Hasil Testing pada Delete Transaksi Penjualan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Pelanggan: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/transaksipenjualan/apotek?KEY=rifki



Gambar 2. 24 Hasil Testing pada Delete Transaksi Penjualan

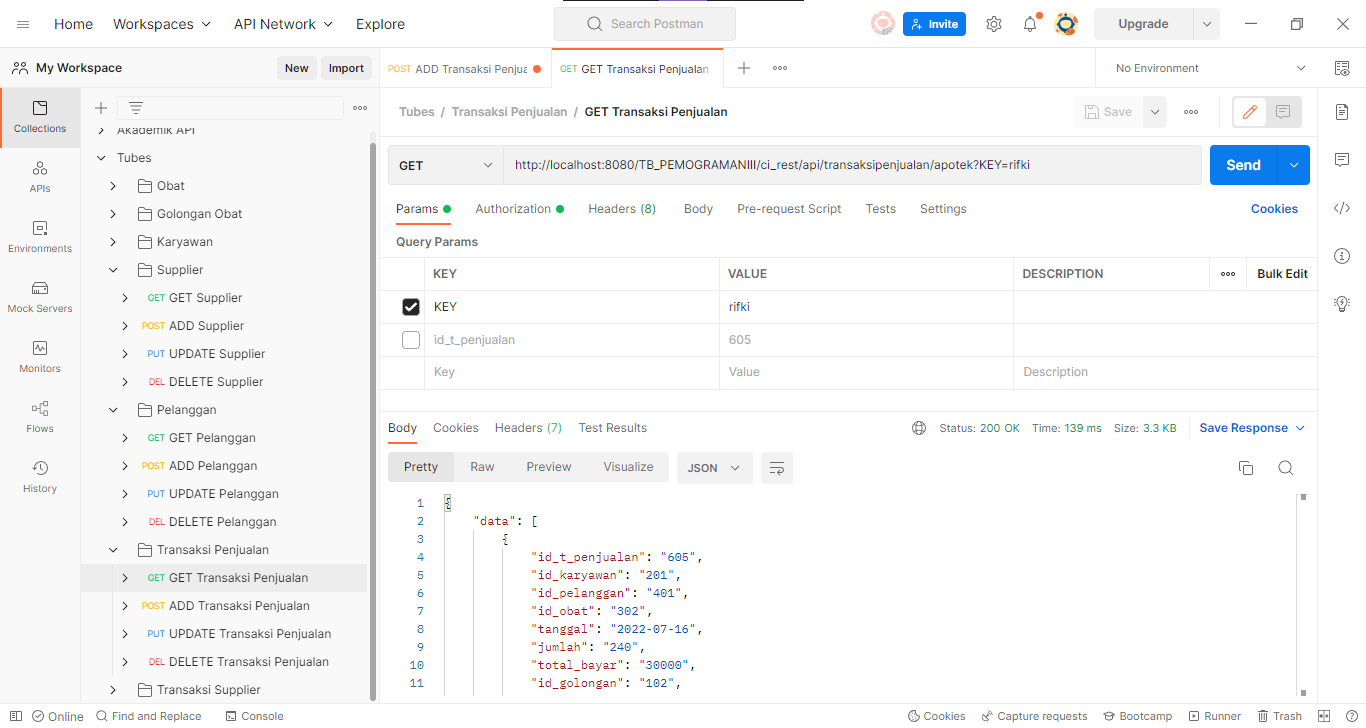
Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Penjualan yang diuji untuk menghapus data Transaksi Penjualan. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Transaksi Penjualan, metode yang digunakan untuk menghapus data Transaksi Penjualan adalah dengan metode POST.

## Endpoint dan Hasil Testing pada Rest Server Transaksi Penjualan

### **Hasil Testing pada GET Transaksi Penjualan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menampilkan data Transaksi Penjualan:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/transaksisupplier/apotek?KEY=algies>



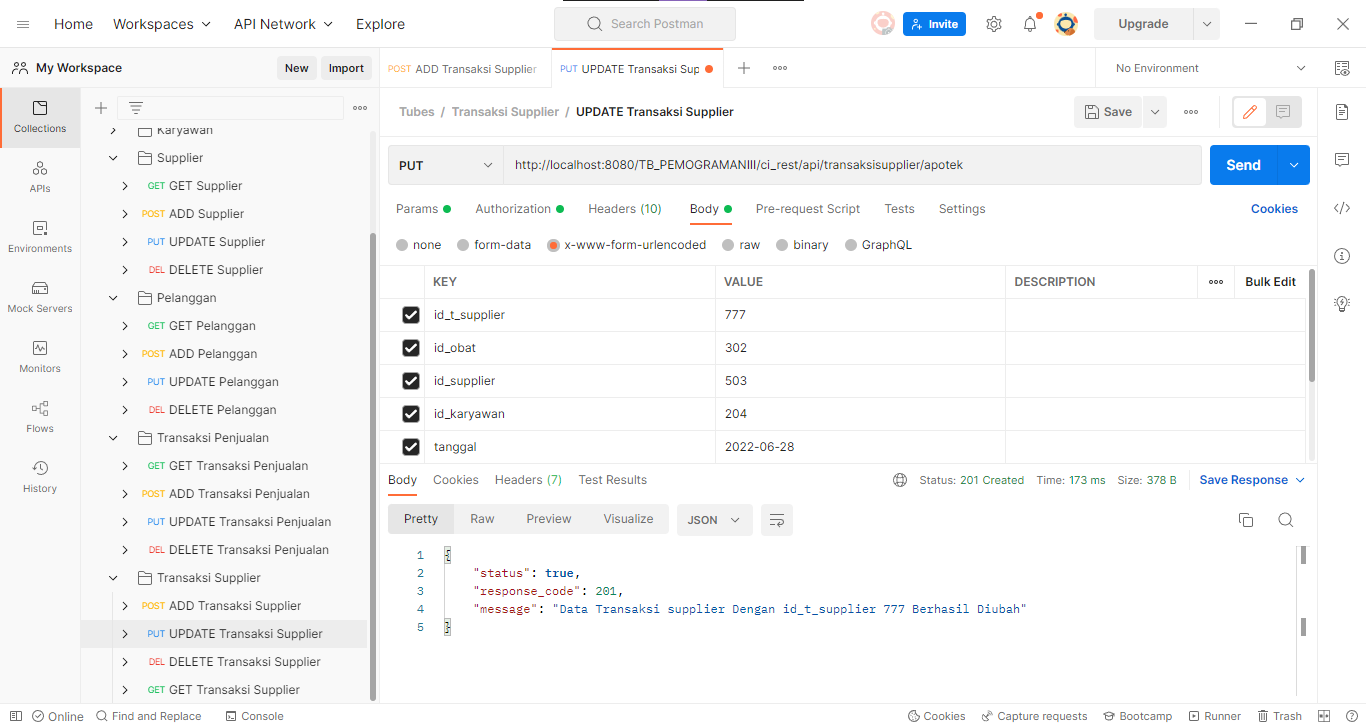
Gambar 2. 25 Hasil Testing pada GET Transaksi Penjualan

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Supplier yang diuji untuk menampilkan seluruh data Transaksi Supplier. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Transaksi Supplier, metode yang digunakan untuk memanggil data Transaksi Supplier adalah dengan metode GET. Dalam menampilkan data Transaksi Supplier, dapat menggunakan Format output data berupa JSON seperti yang terlihat pada gambar.

### **Hasil Testing pada Update Transaksi Supplier**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk mengedit data Transaksi Supplier:

http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/transaksisupplier/apotek



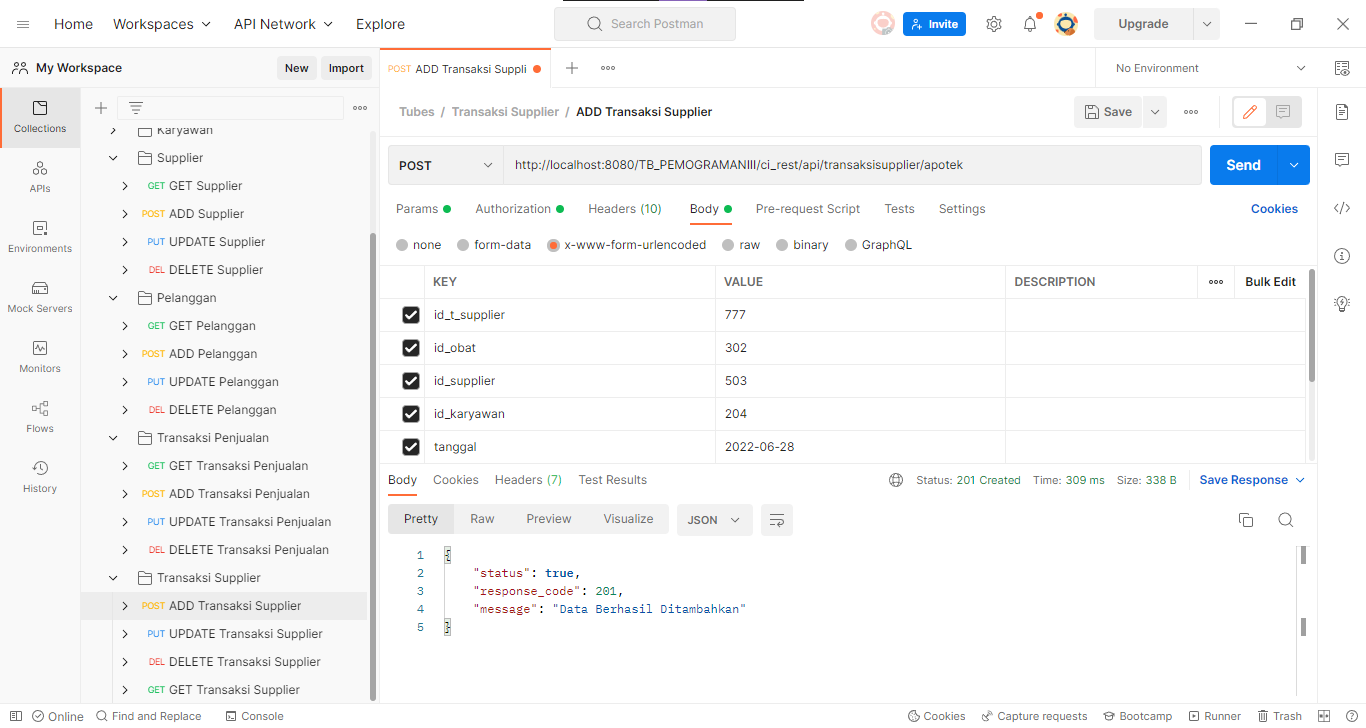
Gambar 2. 26 Hasil Testing pada Update Transaksi Supplier

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Supplier nyang diuji untuk mengedit data Transaksi Supplier. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Transaksi Supplier, metode yang digunakan untuk mengedit data Transaksi Supplier adalah dengan metode PUT.

### **Hasil Testing pada Create**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menambah data Transaksi Supplier:

<http://localhost:8080/TB_PEMOGRAMANIII/ci_rest/api/transaksisupplier/apotek>

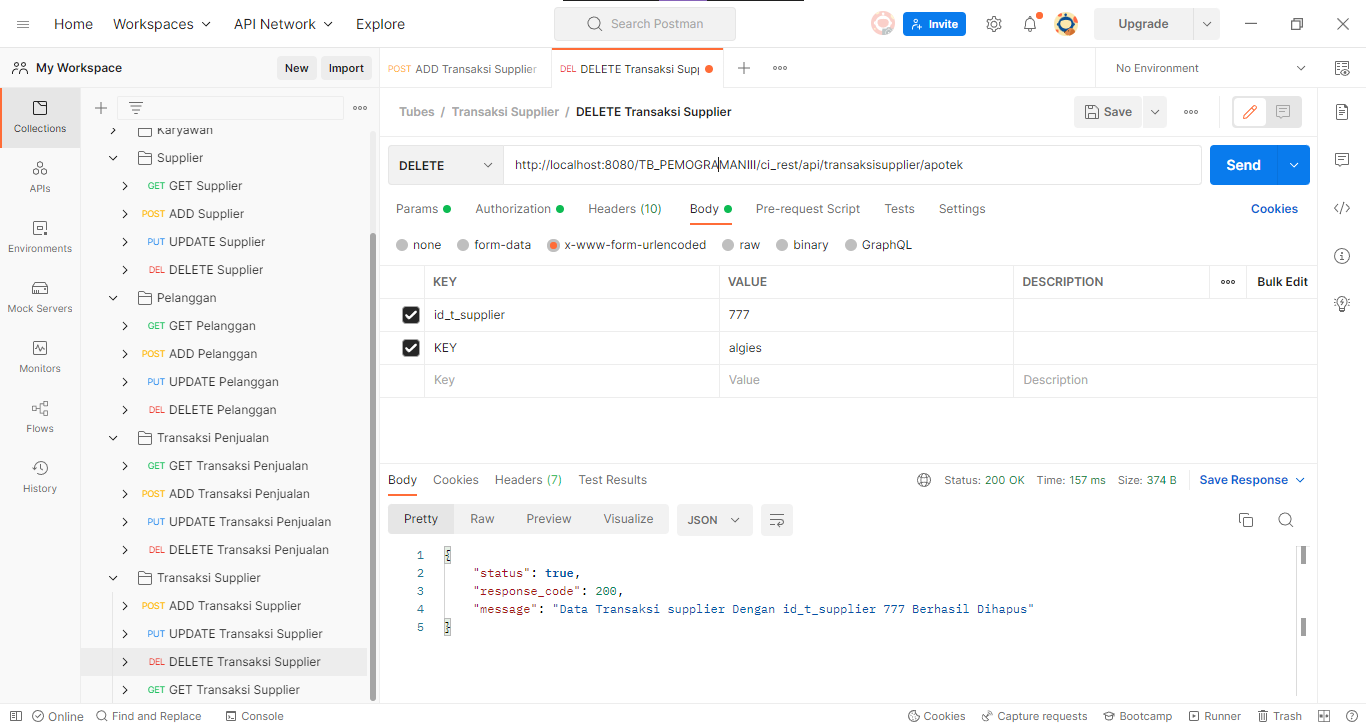


Gambar 2. 27 Hasil Testing pada Create

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Supplier yang diuji untuk menambah data Transaksi Supplier. Dari hasil pengujian tersebut adalah Created dengan request HTTP berupa Status: 200 Created. Pada Endpoint Transaksi Supplier, metode yang digunakan untuk menambah data Transaksi Supplier adalah dengan metode POST.

### **Hasil Testing pada Delete Transaksi Penjualan**

Berikut Single Endpoint yang digunakan dan hasil pengujian untuk menghapus data Pelanggan: http://localhost:8080/TB\_PEMOGRAMANIII/ci\_rest/api/transaksisupplier/apotek

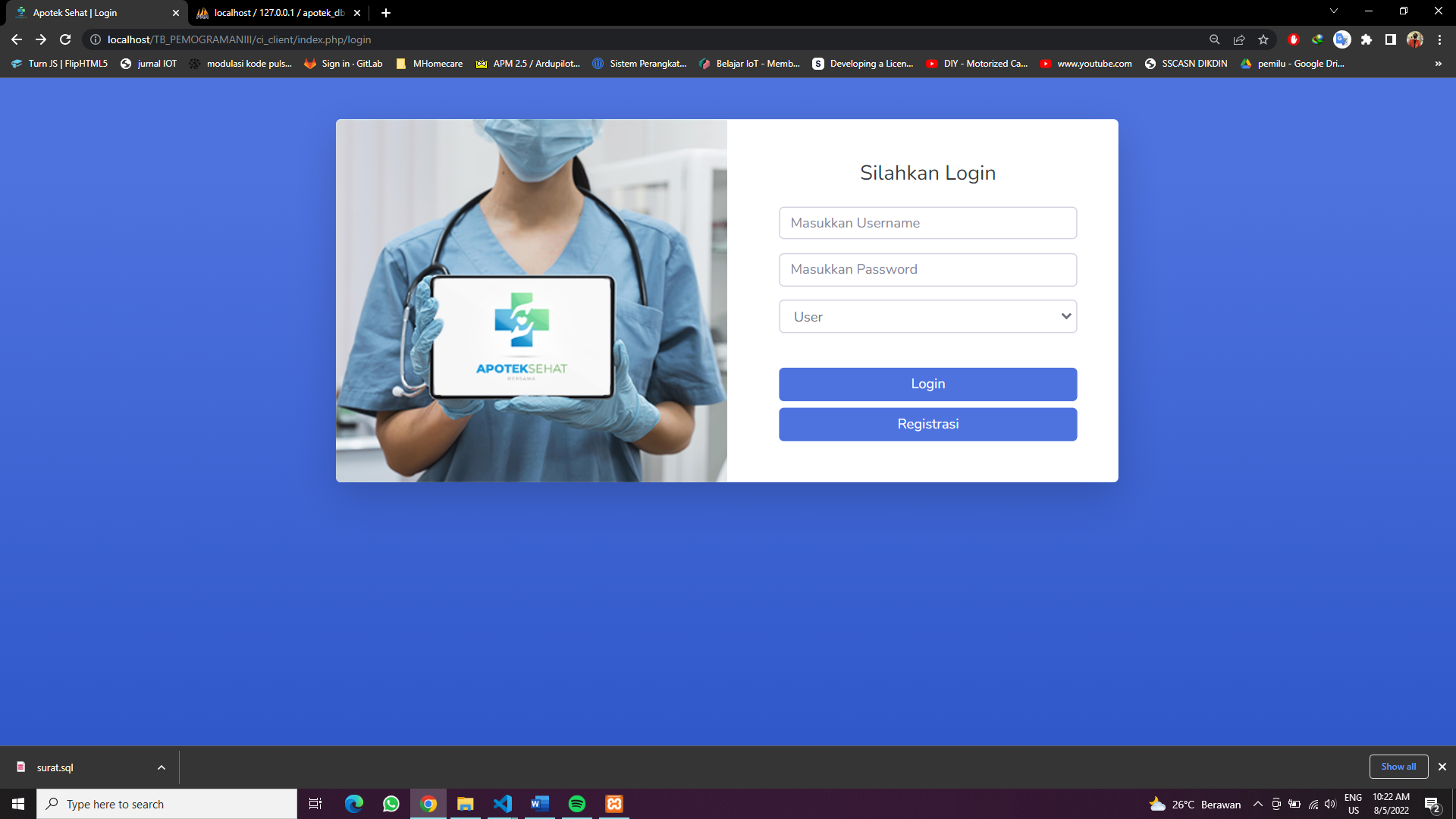


Gambar 2. 28 Hasil Testing pada Delete Transaksi Penjualan

Pada gambar diatas, dapat dilihat mengenai Endpoint Transaksi Supplier yang diuji untuk menghapus data Transaksi Supplier. Dari hasil pengujian tersebut adalah OK dengan request HTTP berupa Status: 200 OK. Pada Endpoint Transaksi Supplier, metode yang digunakan untuk menghapus data Transaksi Supplier adalah dengan metode POST.

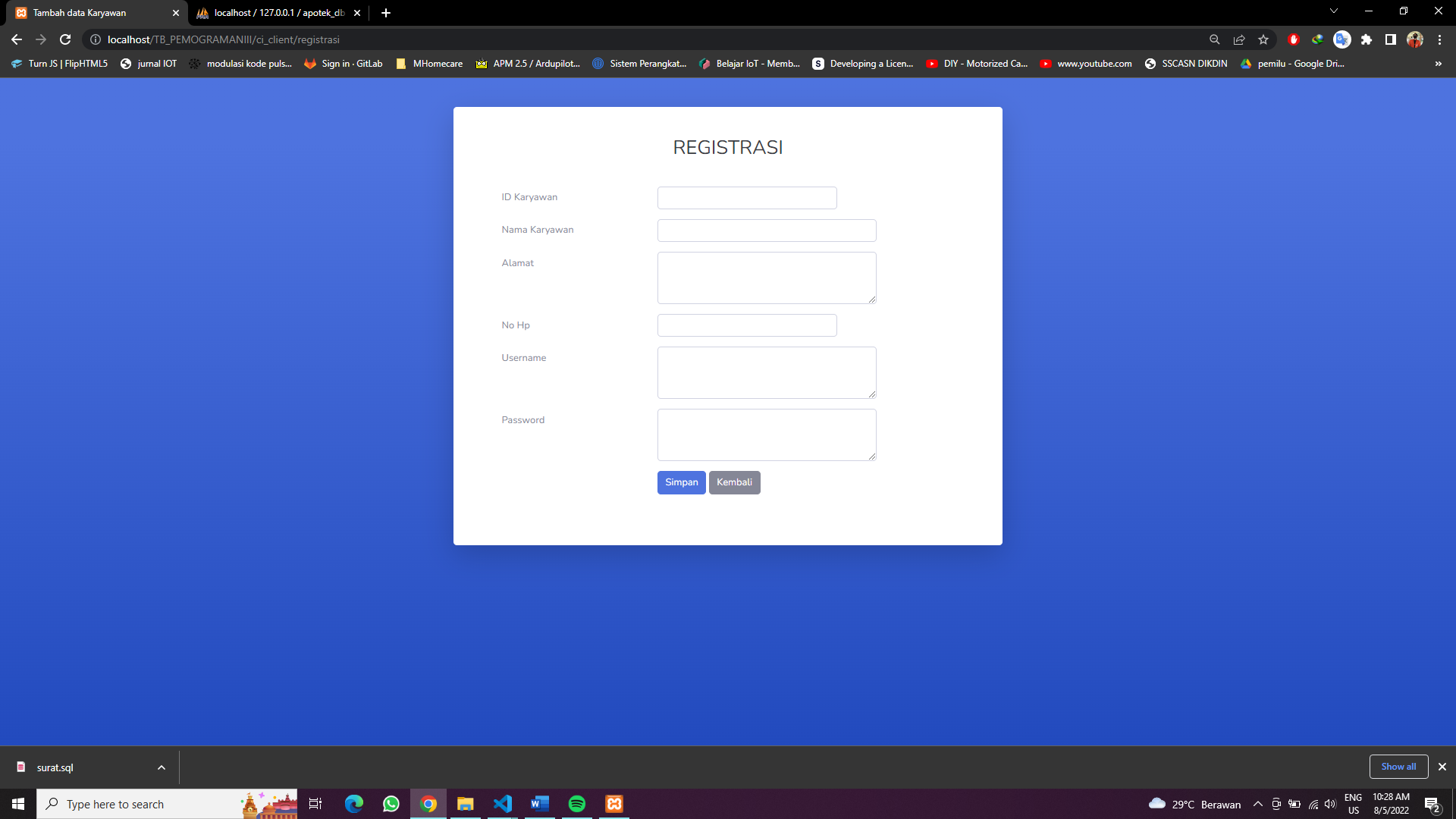
# Output untuk masing-masing endpoint

## Tampilan Output dari View Login



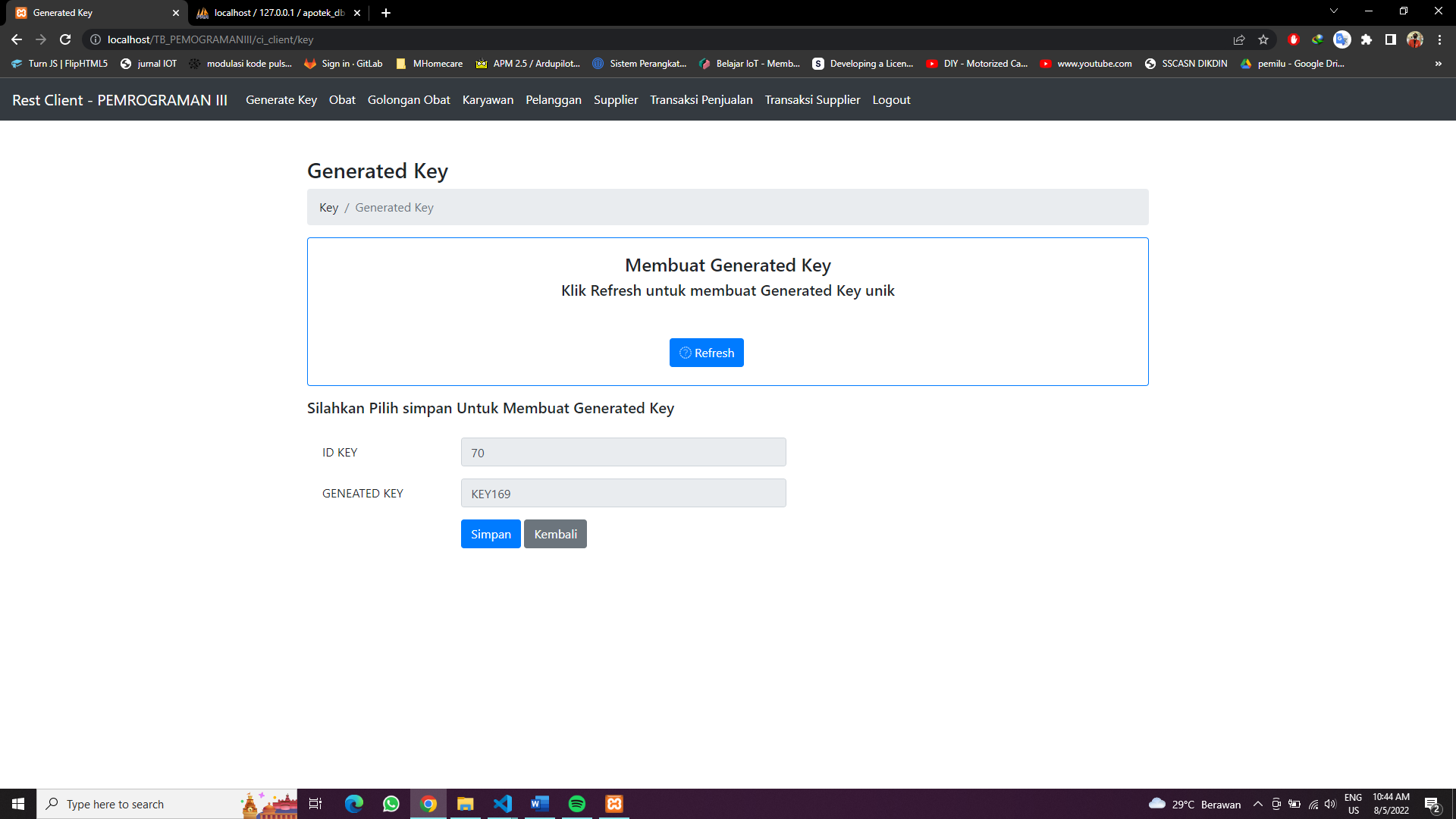
Pada gambar diatas, dapat dilihat mengenai tampilan login, dapat memasukan username dan password pada form yang sudah disediakan, sebagai user masuk kelogin, jika belum memiliki akun maka dapat mendaftar/register terlebih dahulu pada tombol dibawah login.

## Tampilan Output dari View Register

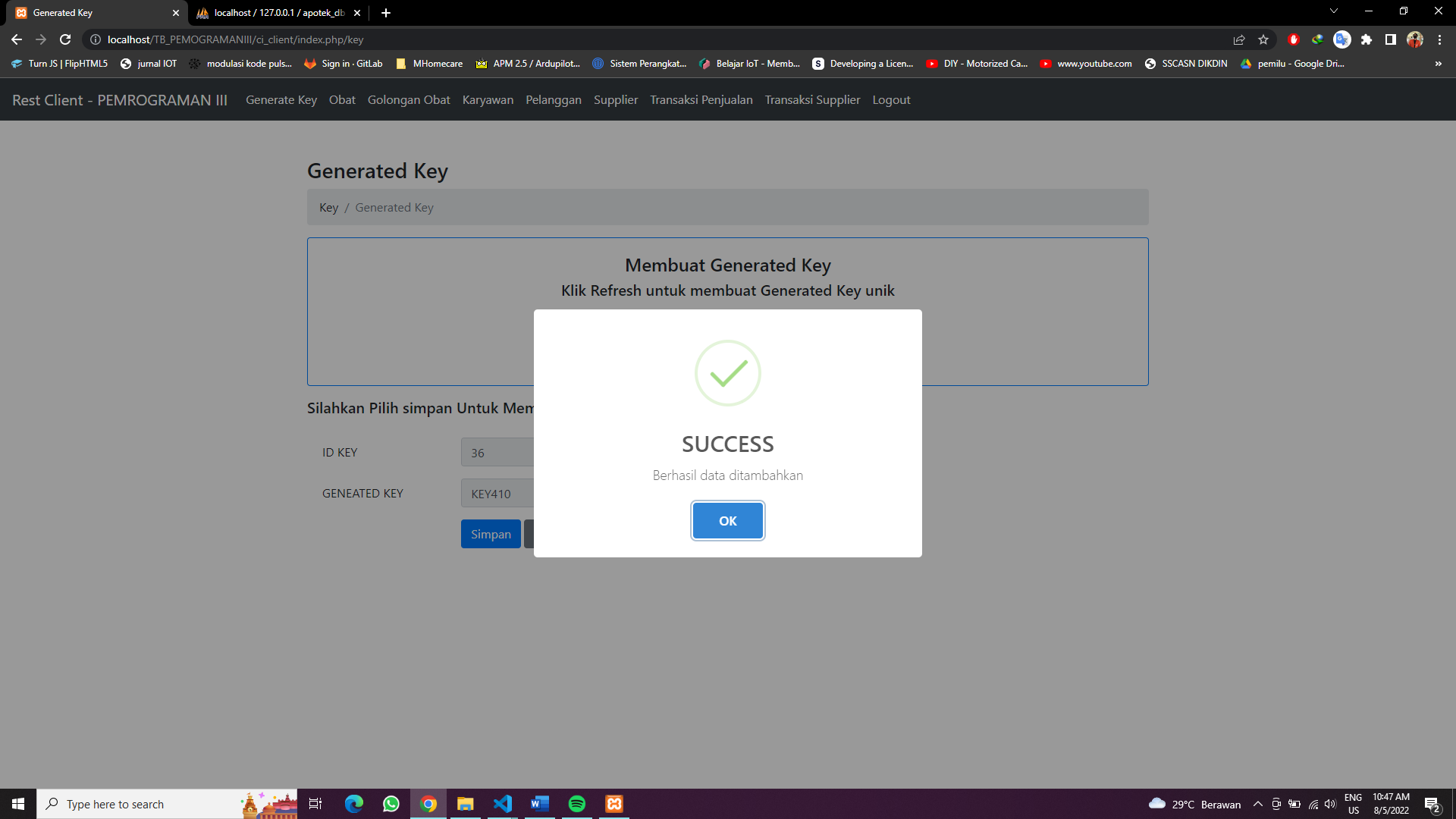


Pada tampilan registrasi diatas, dapat dilihat untuk registrasi harus dapat memenuhi dengan memasukan ID Karyawan, Nama Karyawan, Alamat, No Hp, Username dan Password pada form yang sudah disediakan, sebagai user register untuk bisa masuk ke login, jika sudah maka bisa menekan tombol “Simpan”, dan tombol “kembali” untuk kembali ke halaman login jika tidak ingin/batal melakukan registrasi.

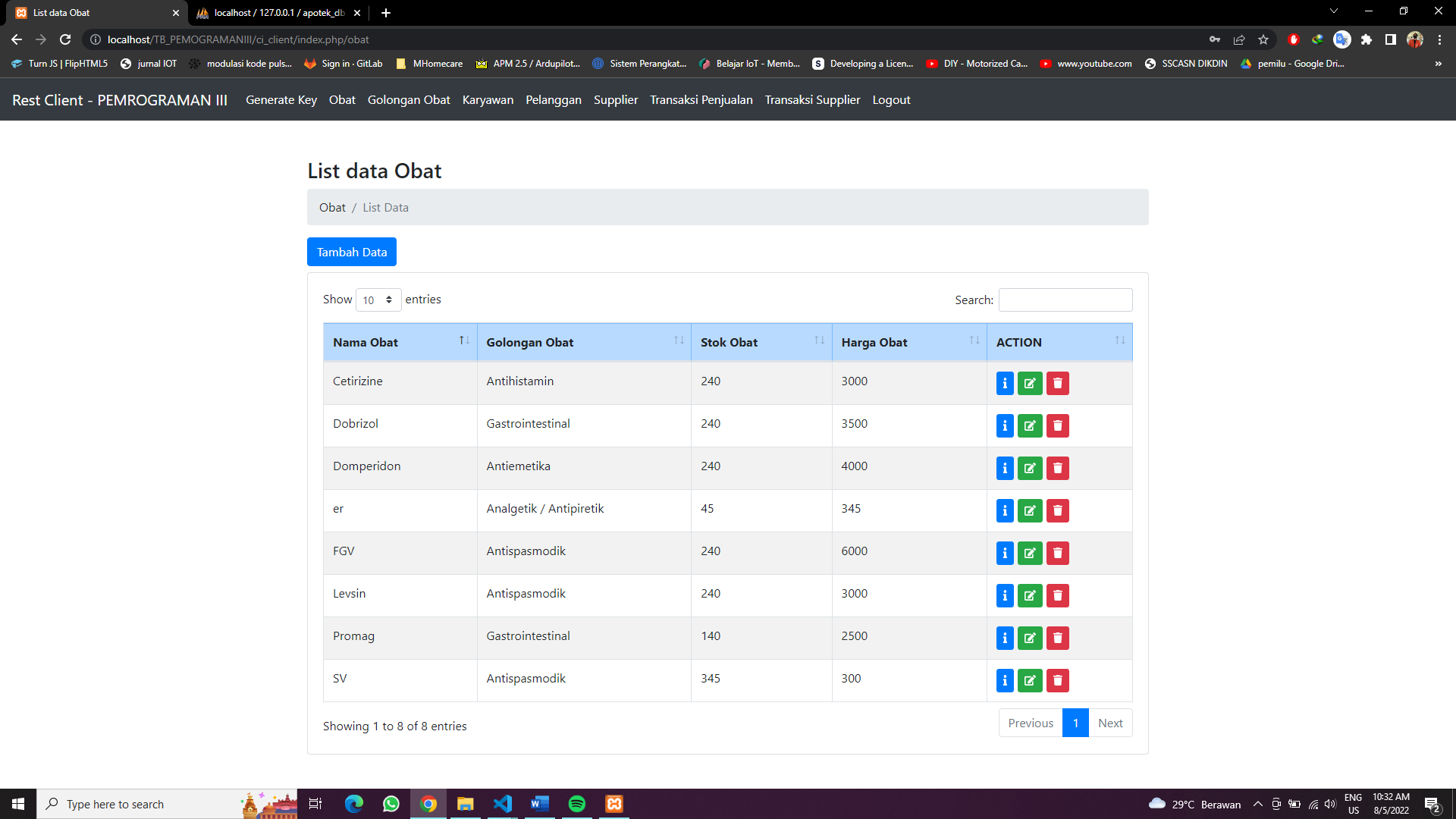
## Tampilan Output dari View Generated Key



Pada tampilan Generated key, dapat menekan tombol refresh untuk mendapatkan IDKey dan Generated key secara acak, klik simpan hingga pop-up berhasil muncul



## Tampilan Output dari View Data Obat



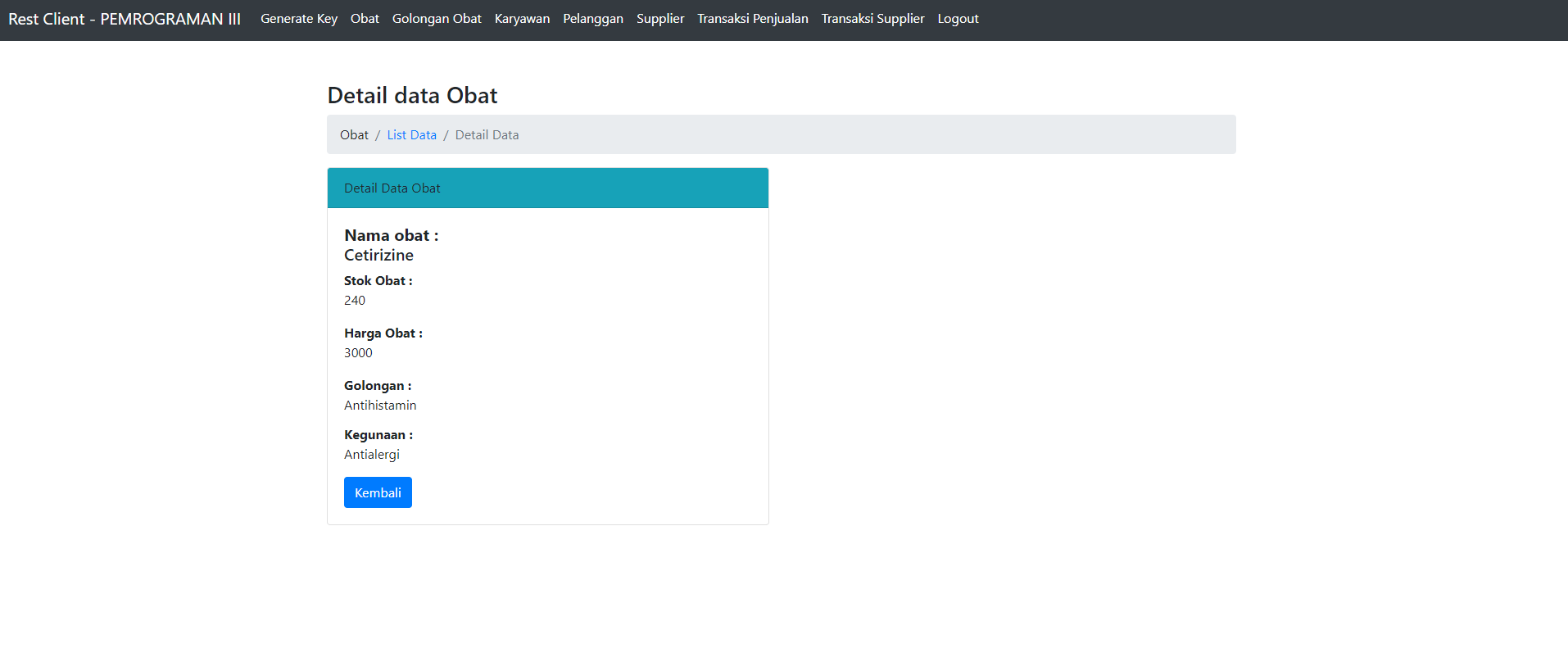
Setelah login maka akan muncul list data obat yang dapat ditambah, edit, dan delete. Dan tampilan obat diatas, dapat dilihat ada beberapa macam obat yang sudah ditambahkan, jika ingin menambahkan obat dapat menekan tombol Tambah obat dan jika ingin melihat keterangan obat, mengedit, dan menghapus bisa menekan tombol action pada bagian kanan tabel.

### **View Tambah data obat**



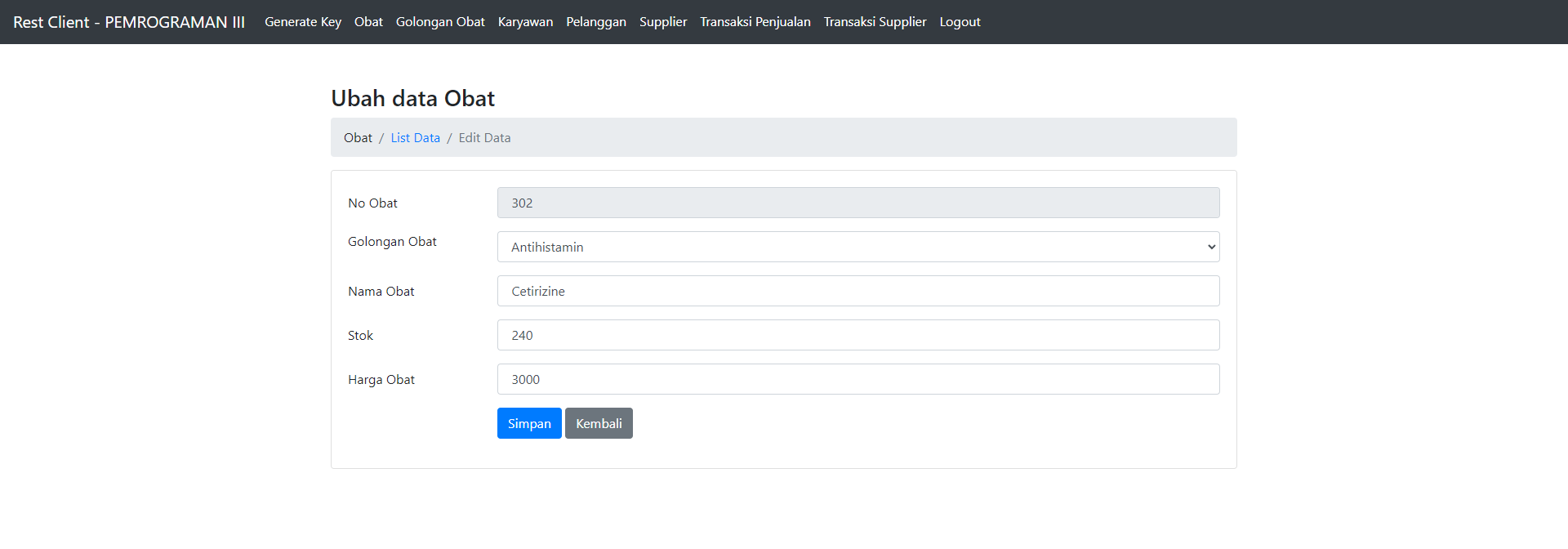
Jika sudah menekan tombol tambah obat maka akan muncul tampilan obat diatas, dapat dilihat untuk menambahkan obat kita harus memenuhi form dari obat yang ingin ditambahkan seperti no, nama, golongan, stok, harga dan klik simpan untuk menyimpan obat yang ingin ditambahkan, tombol kembali untuk kembali ke halaman obat jika batal menambahkan obat.

### **View List data obat**



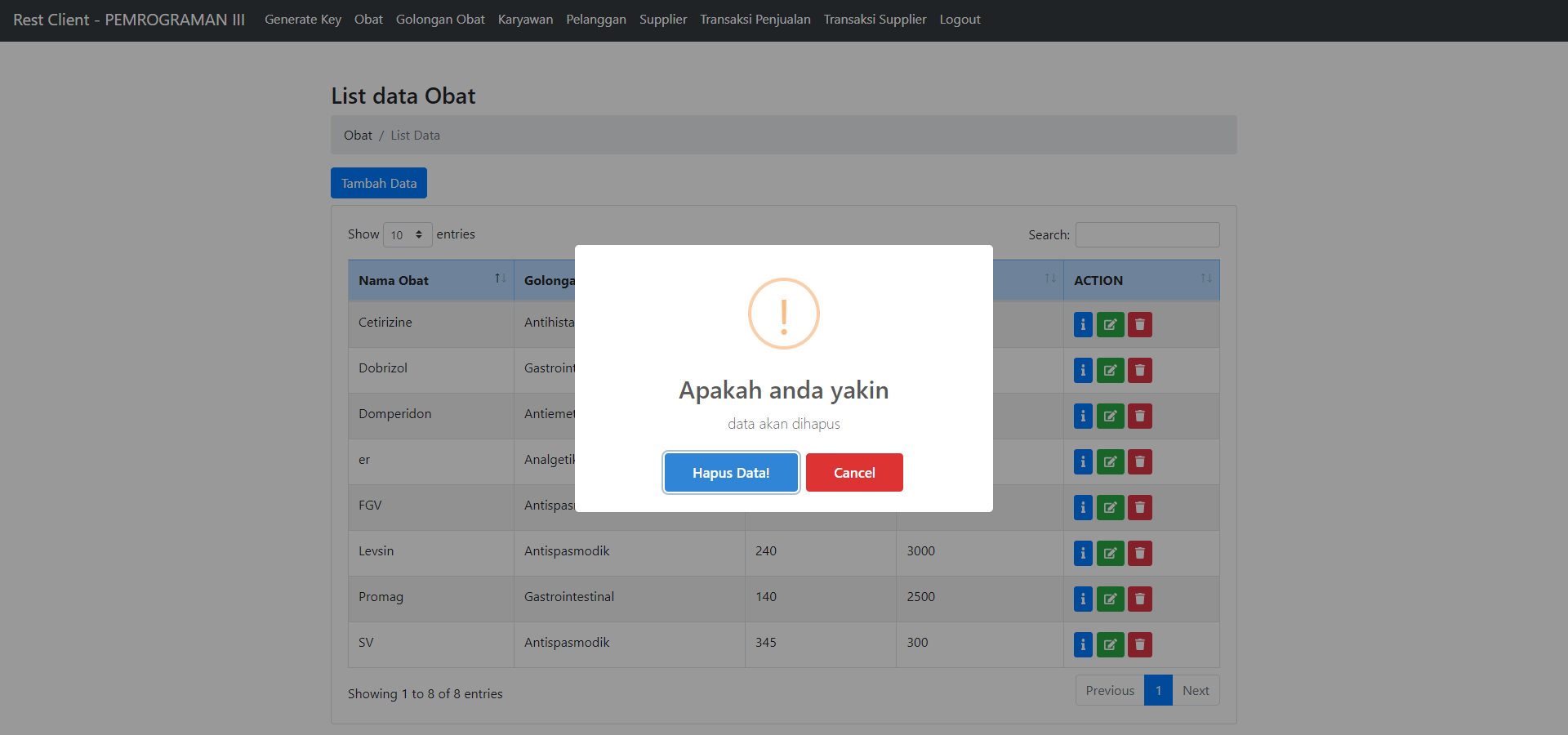
Jika sudah menekan tombol tambah info obat berwarna biru pada action maka akan muncul tampilan list obat seperti ini.

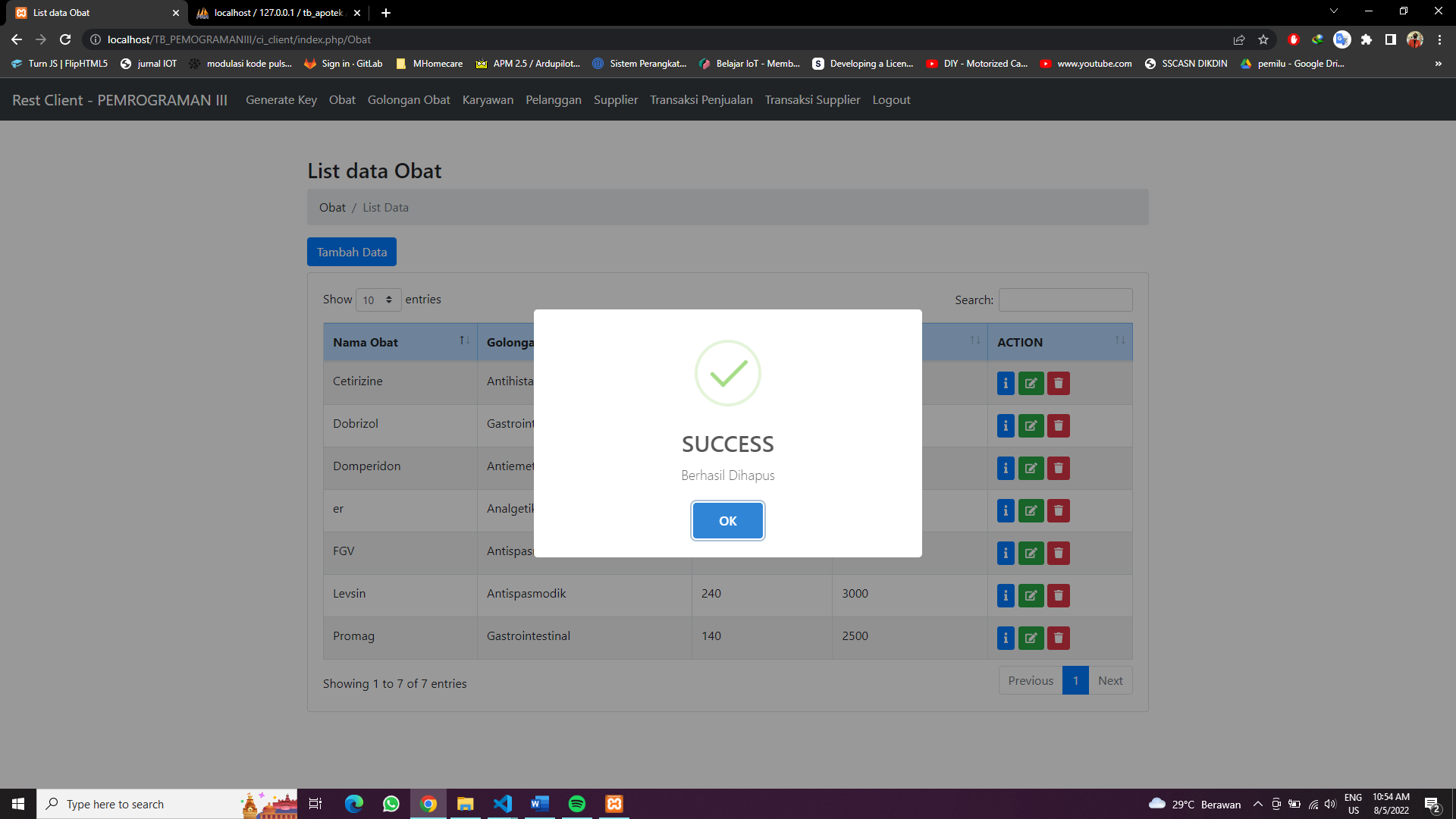
### **View Ubah/Edit data obat**



Jika sudah menekan tombol tambah edit data obat berwarna hijau pada action maka akan muncul tampilan edit obat seperti ini.

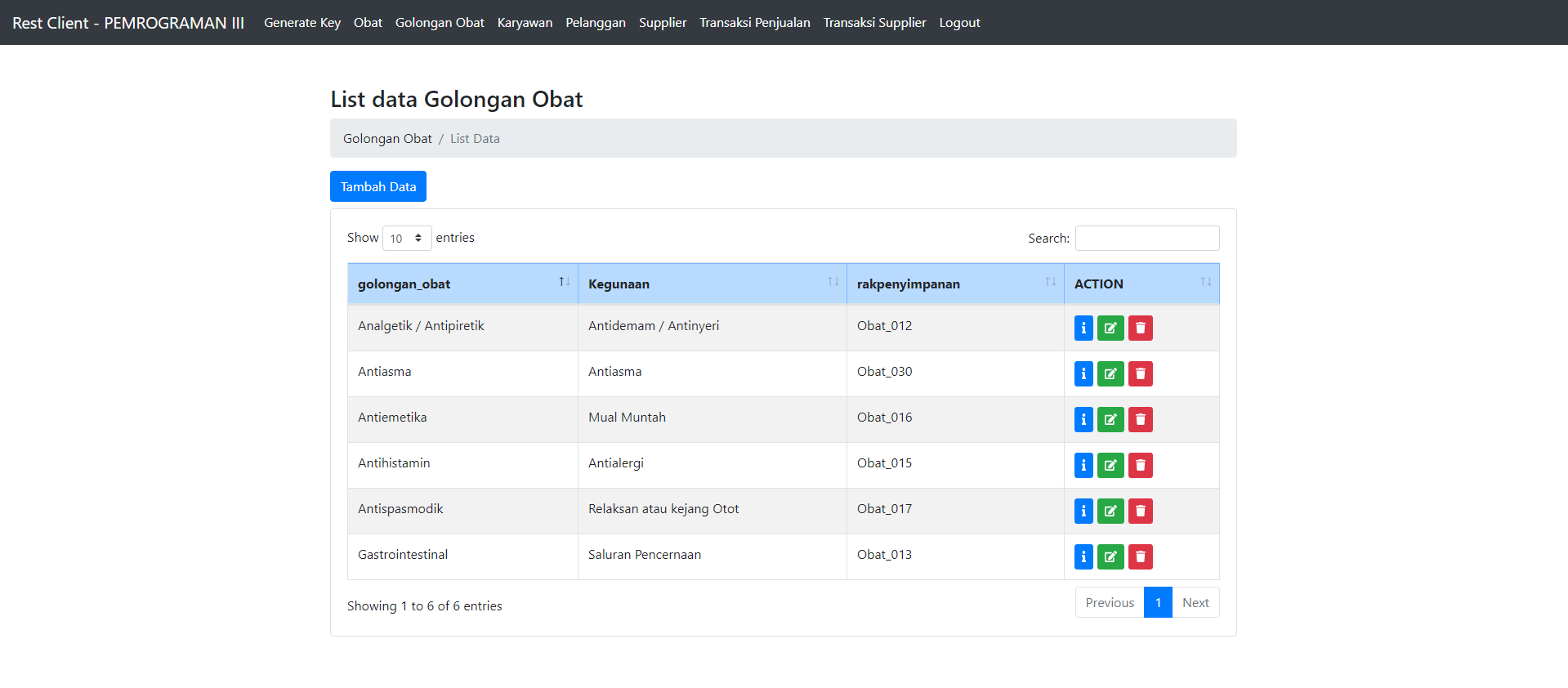
### **View Delete/Hapus data obat**





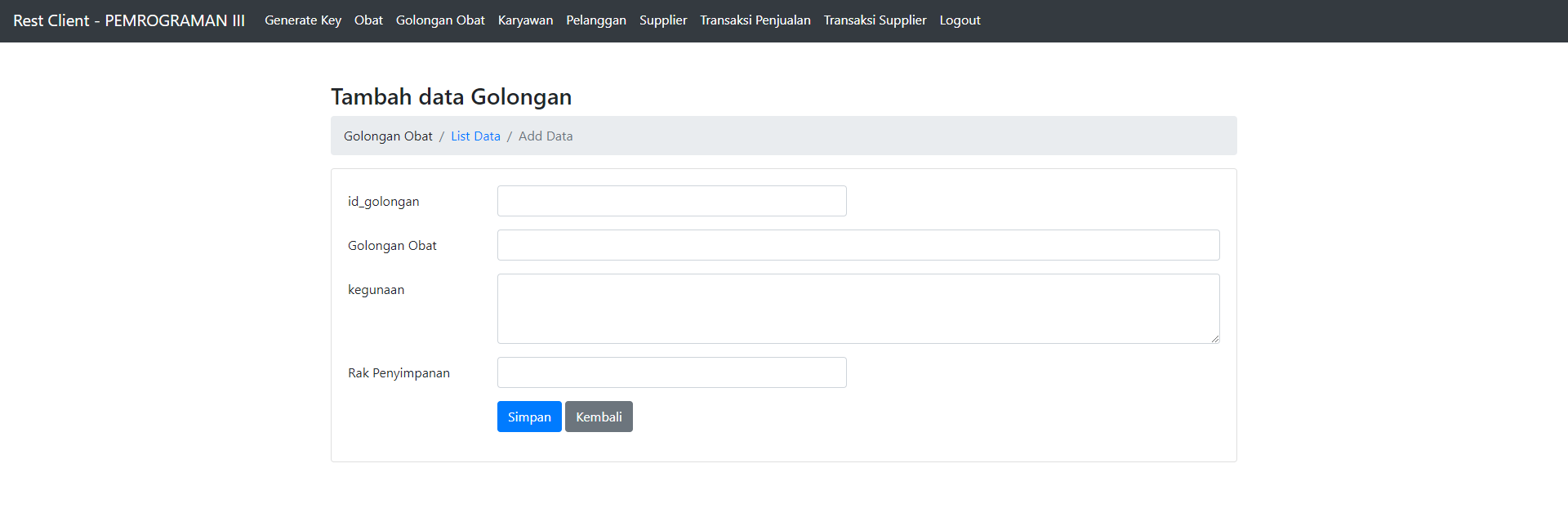
Jika sudah menekan tombol hapus data obat berwarna merah pada action maka akan muncul tampilan hapusnya akan seperti ini.

## Tampilan Output dari View Golongan Obat



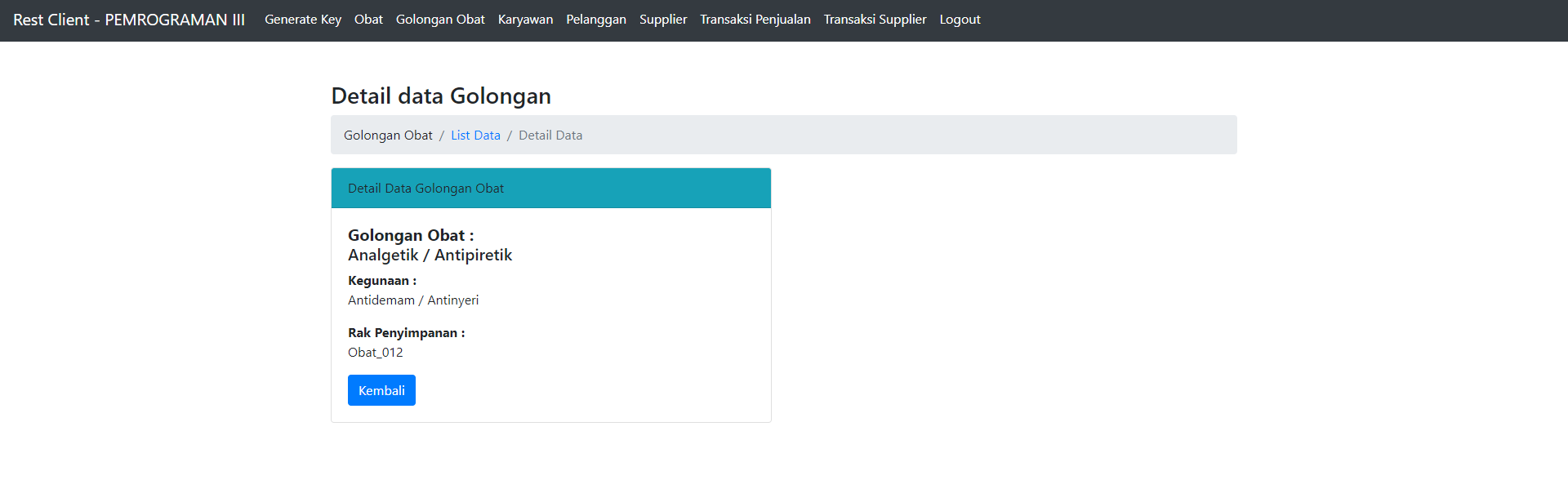
Dapat dilihat ada beberapa macam golongan obat yang sudah ditambahkan, jika ingin menambahkan obat dapat menekan tombol Tambah golongan obat dan jika ingin melihat keterangan golongan obat, mengedit, dan menghapus bisa menekan tombol action pada bagian kanan tabel.

### **View Tambah data golongan obat**



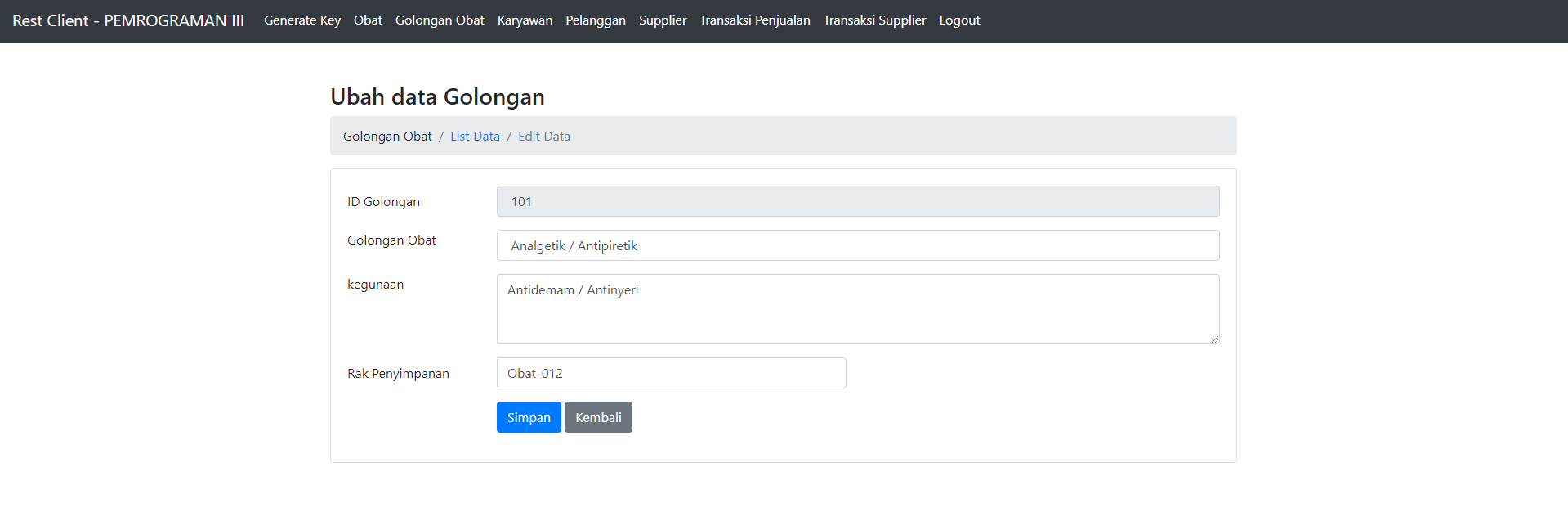
Jika sudah menekan tombol tambah golongan obat maka akan muncul tampilan golongan obat diatas, dapat dilihat untuk menambahkan kita harus memenuhi form dari obat yang ingin ditambahkan seperti id\_golongan, golongan obat, kegunaan, rak penyimpanan dan klik simpan untuk menyimpan obat yang ingin ditambahkan, tombol kembali untuk kembali ke halaman obat jika batal menambahkan golongan obat.

### **View List data golongan obat**



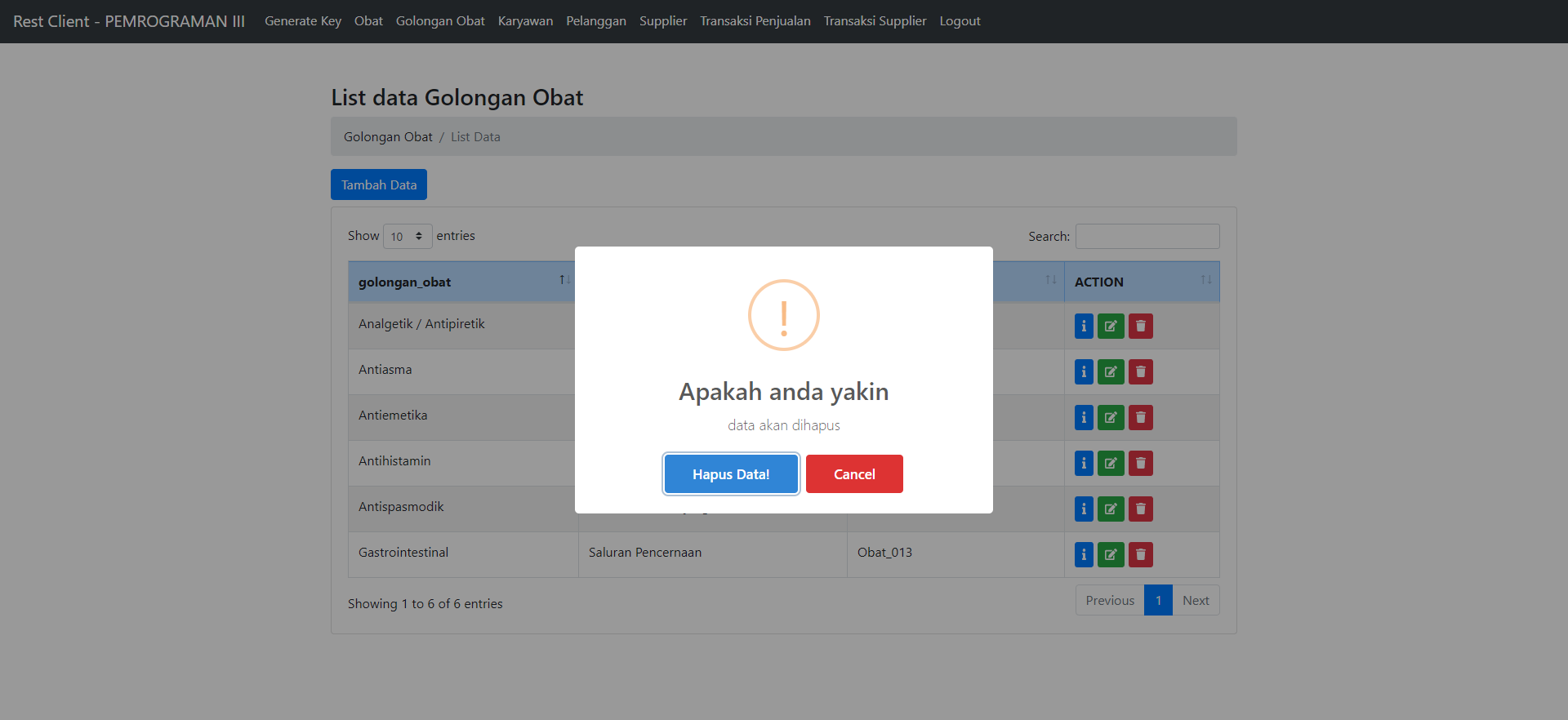
Jika sudah menekan tombol tambah info obat berwarna biru pada action maka akan muncul tampilan list golongan obat seperti ini.

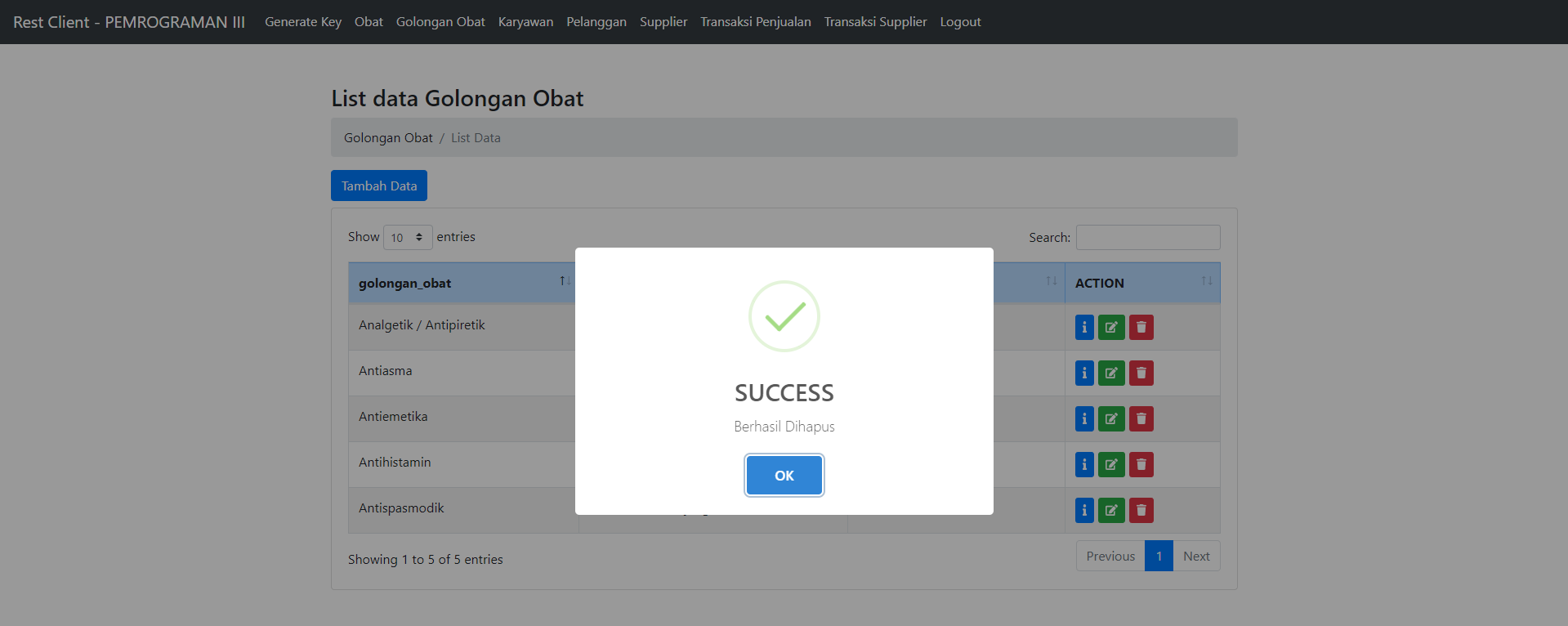
### **View Ubah/Edit data golongan obat**



Jika sudah menekan tombol tambah edit data obat berwarna hijau pada action maka akan muncul tampilan edit gologan obat seperti ini.

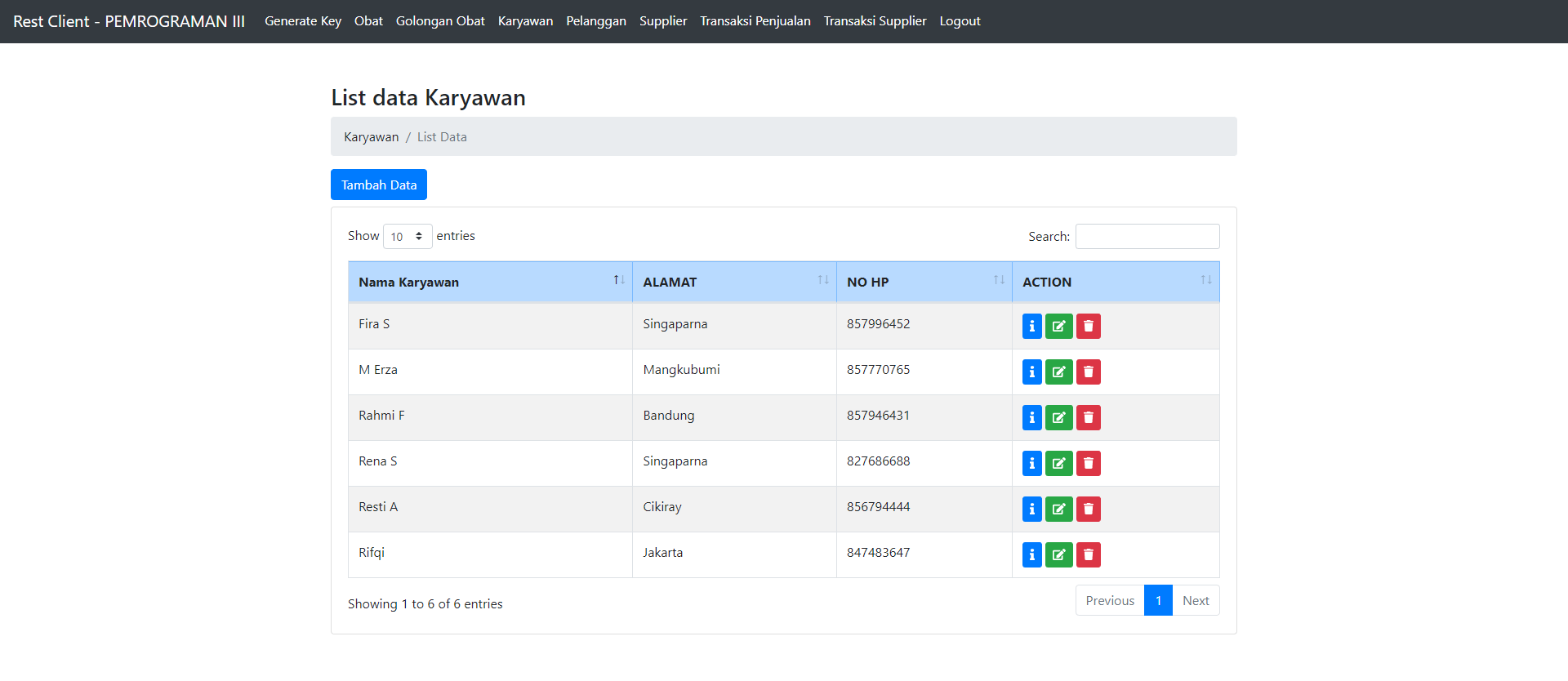
### **View Delete/Hapus data golongan obat**





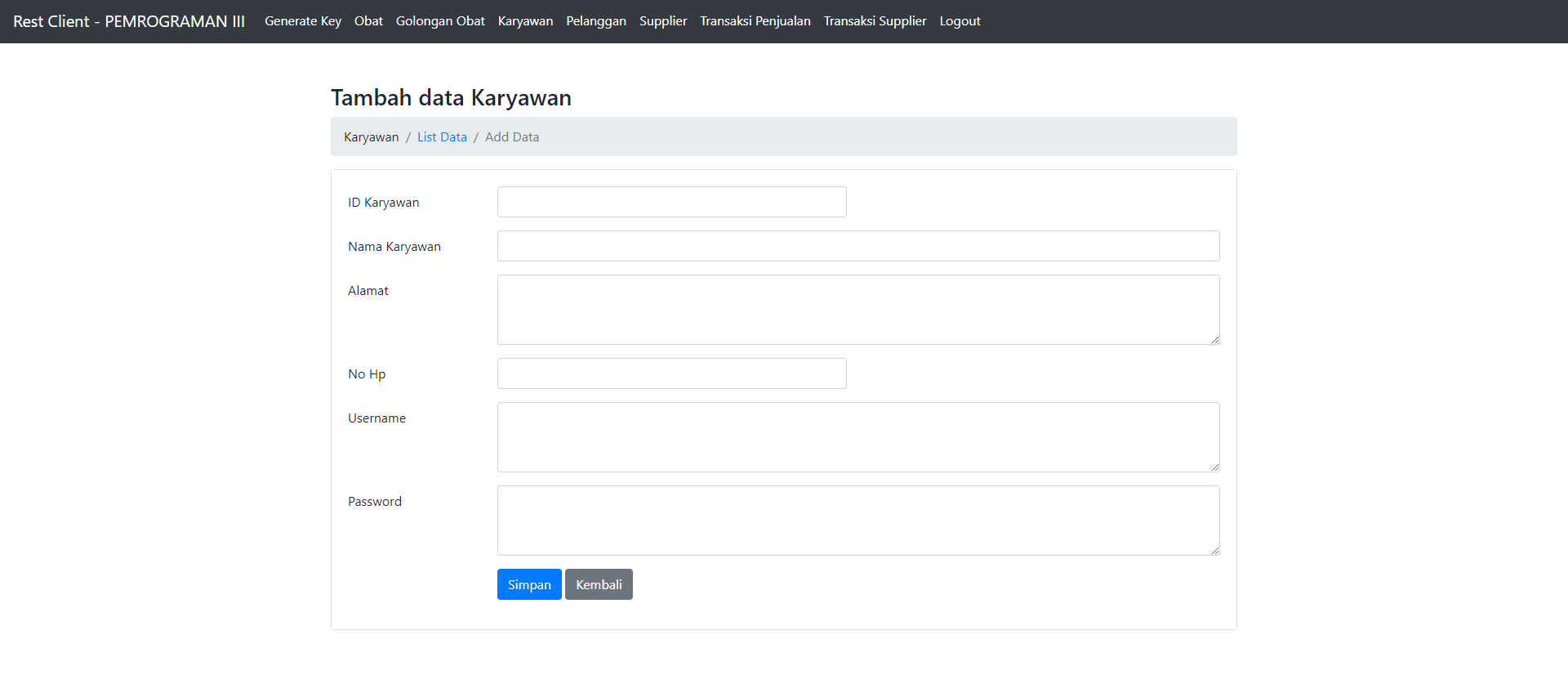
Jika sudah menekan tombol hapus data golongan obat berwarna merah pada action maka akan muncul tampilan hapusnya akan seperti ini.

## Tampilan Output dari View Karyawan Apotek



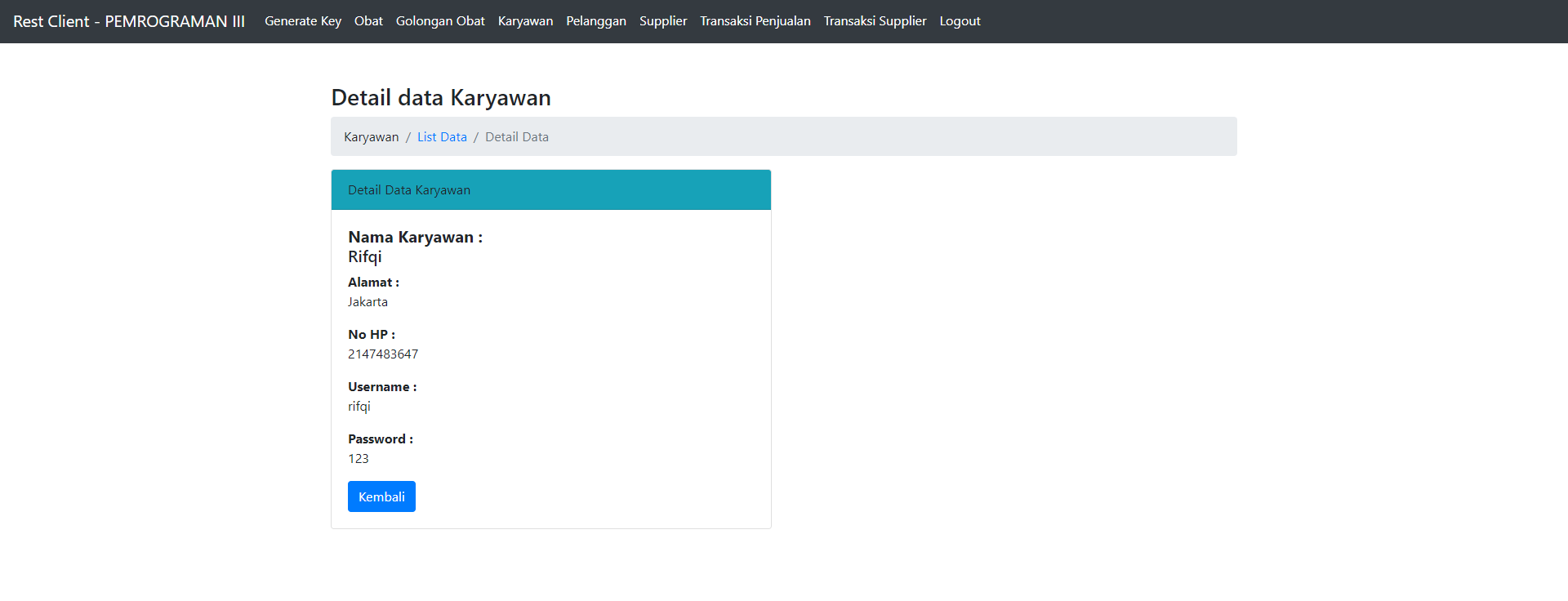
Dapat dilihat ada beberapa karyawan yang sudah ditambahkan, jika ingin menambahkan karyawan dapat menekan tombol Tambah karyawan dan jika ingin melihat keterangan karyawan, mengedit, dan menghapus bisa menekan tombol action pada bagian kanan tabel.

### **View Tambah data karyawan**



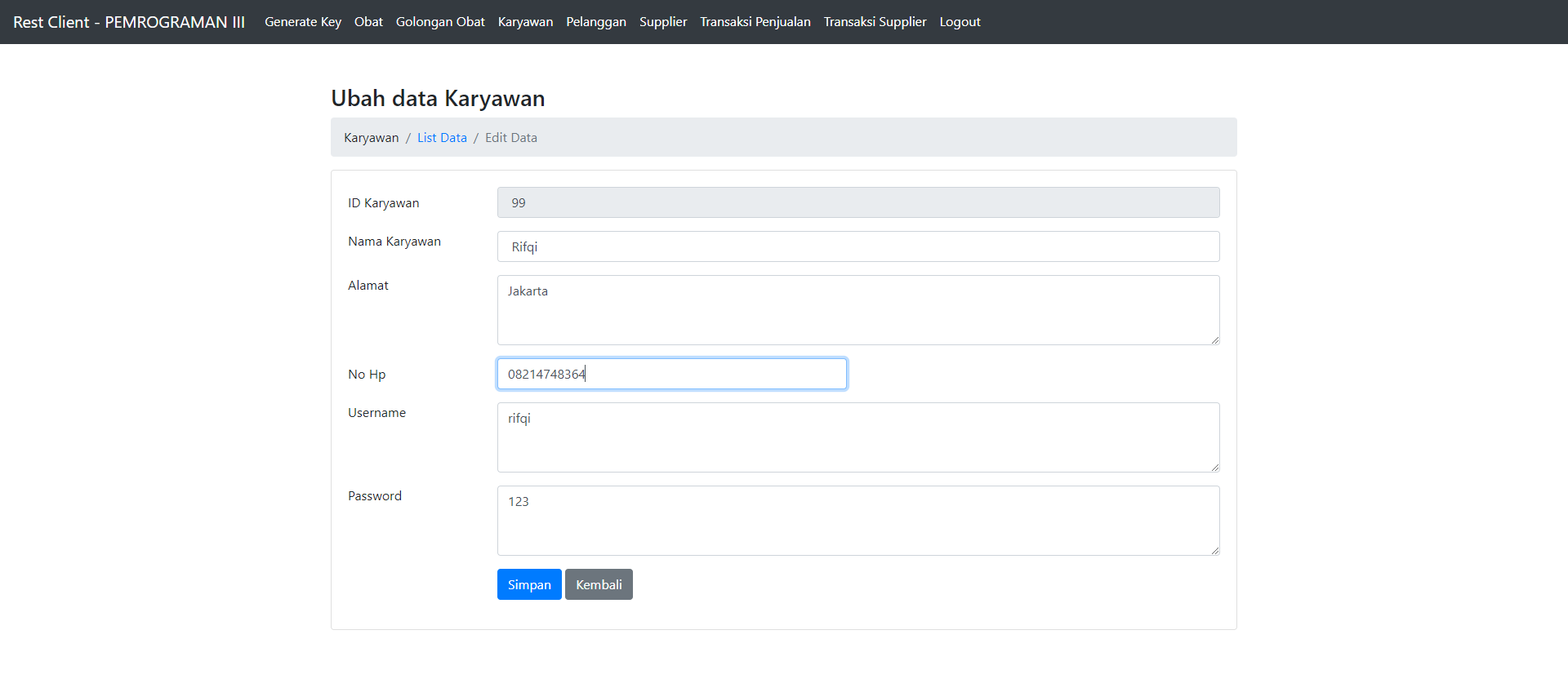
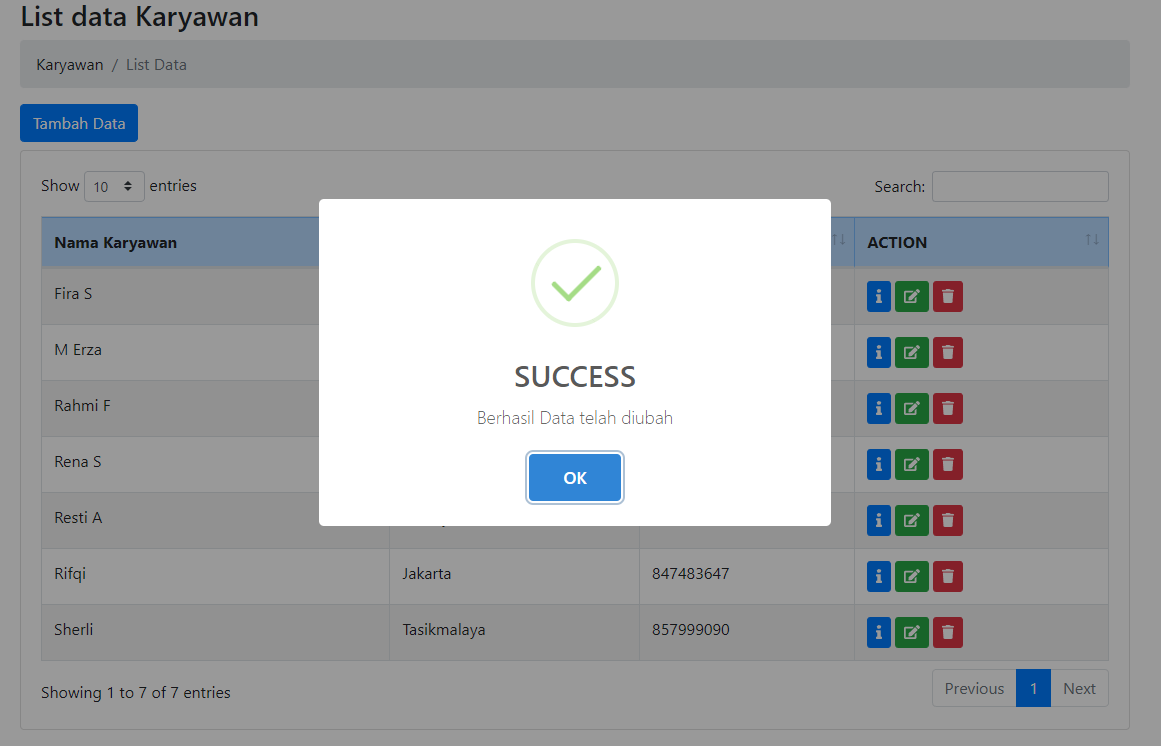
Jika sudah menekan tombol tambah karyawan maka akan muncul tampilan karyawan diatas, dapat dilihat untuk menambahkan kita harus memenuhi form dari pelanggan yang ingin ditambahkan seperti id\_karyawan, nama pelanggan, alamat, no hp, username, password dan klik simpan untuk menyimpan data karyawan yang ingin ditambahkan, tombol kembali untuk kembali ke halaman karyawan jika batal menambahkan karyawan.

### **View List data karyawan**



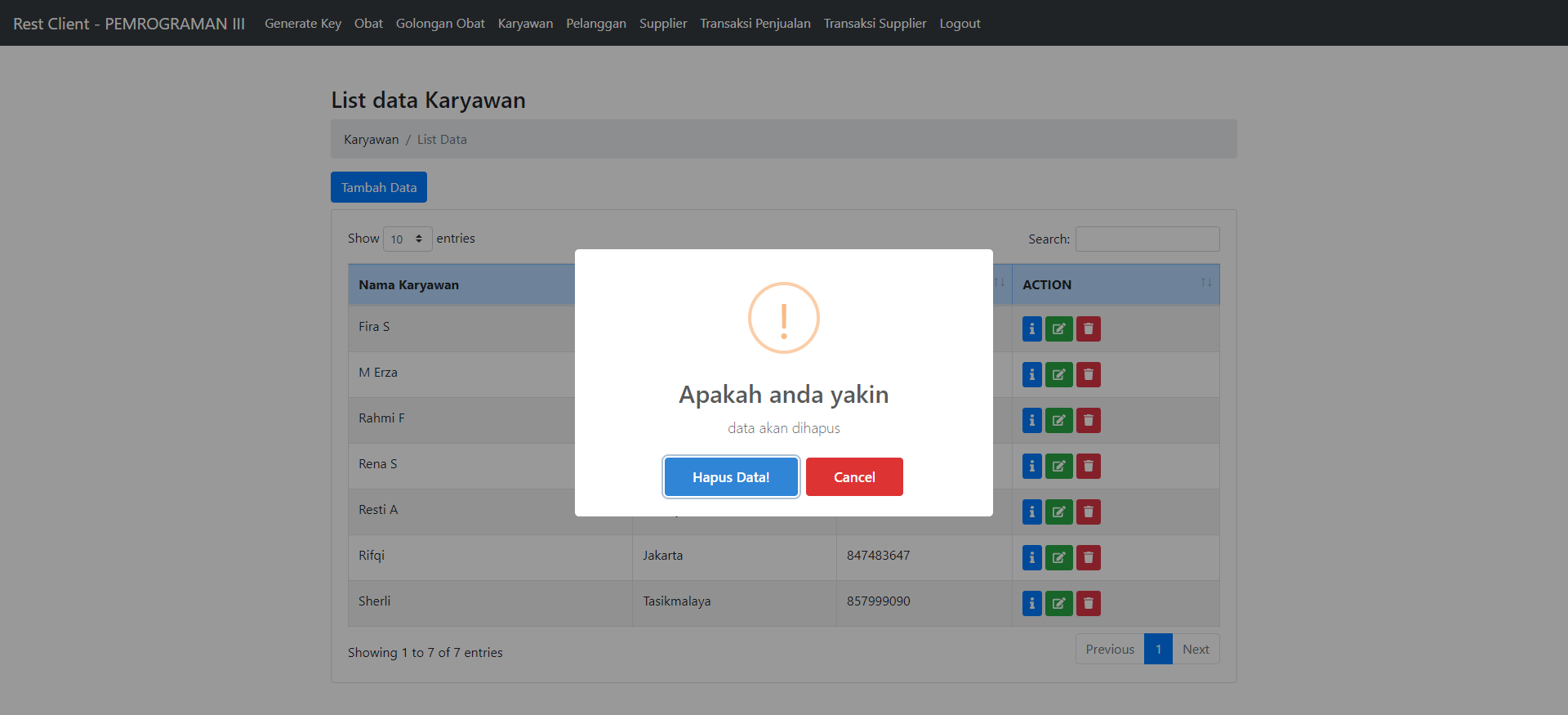
Jika sudah menekan tombol tambah info berwarna biru pada action maka akan muncul tampilan salah satu karyawan yang dipilih seperti ini.

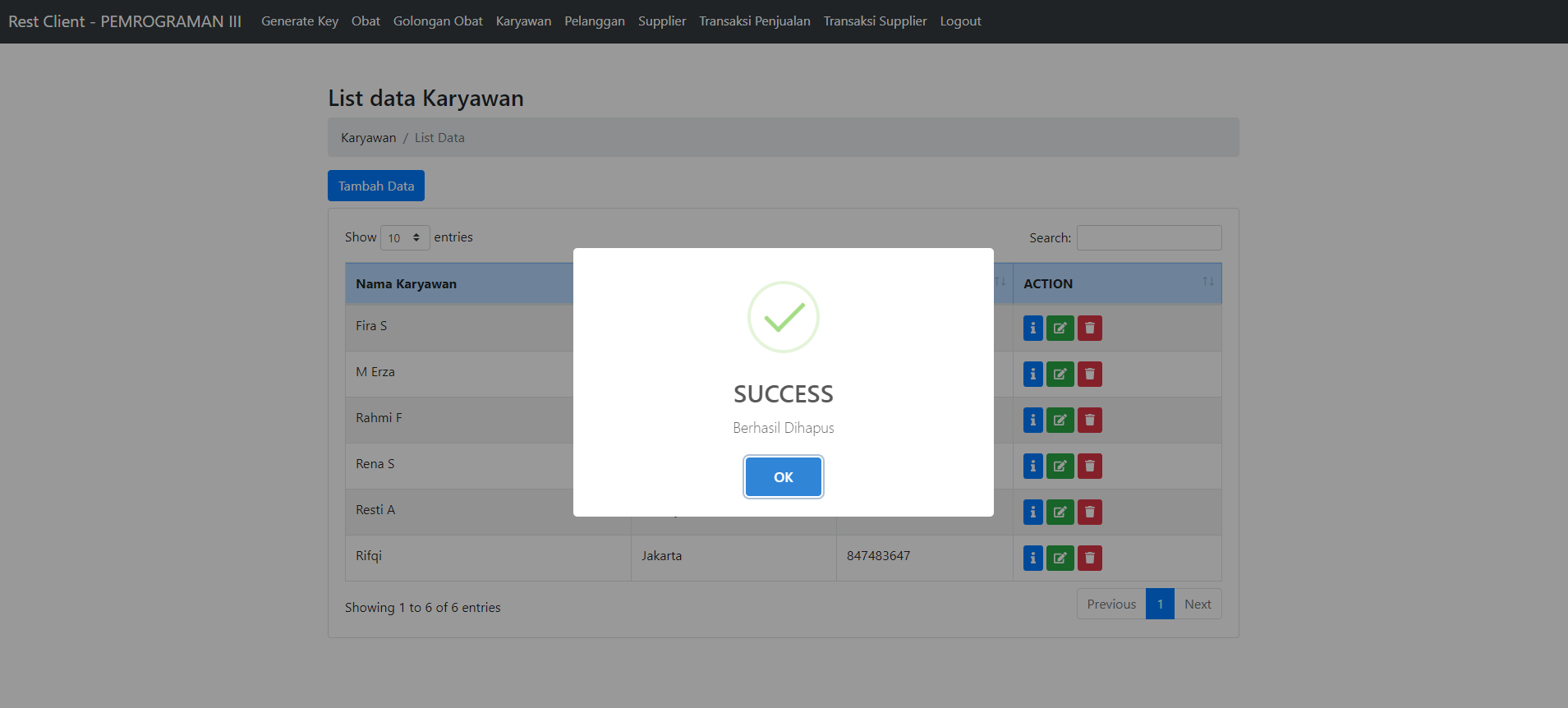
### **View Ubah/Edit data karyawan**

Jika sudah menekan tombol tambah edit karyawan berwarna hijau pada action maka akan muncul tampilan edit simpan karyawan berhasil seperti ini.

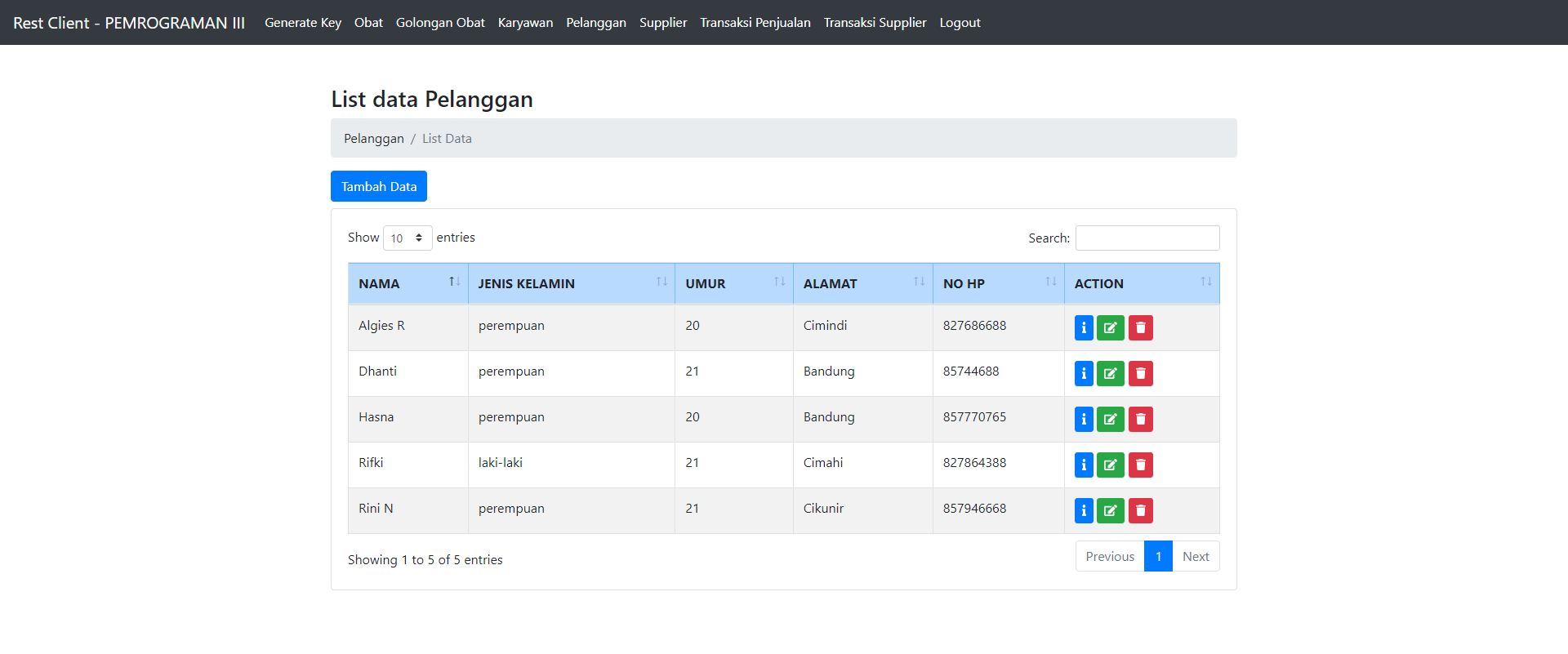
### **View Delete/Hapus data karyawan**





Jika sudah menekan tombol hapus data karyawan berwarna merah pada action maka akan muncul tampilan hapus berhasil akan seperti ini.

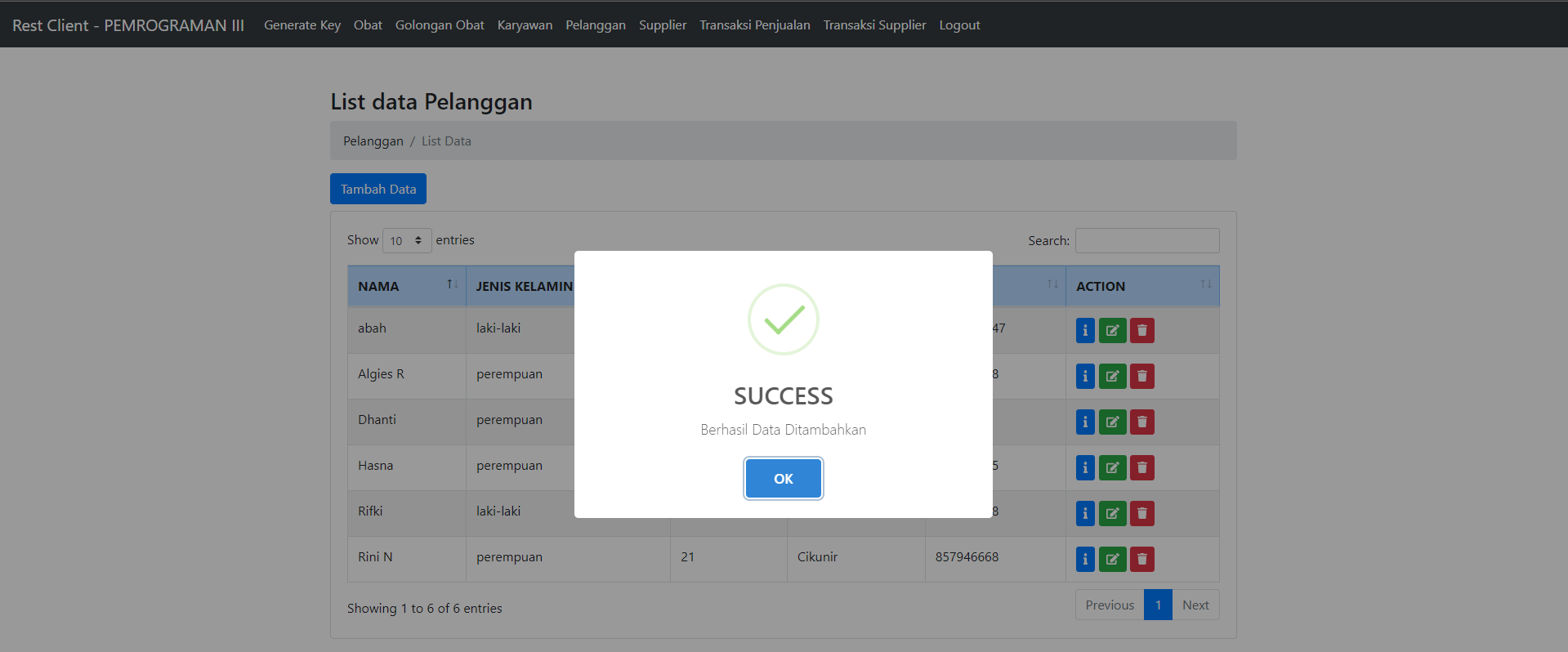
## Tampilan Output dari View Pelanggan Apotek



Dapat dilihat ada beberapa pelanggan yang sudah ditambahkan, jika ingin menambahkan pelanggan dapat menekan tombol Tambah pelanggan dan jika ingin melihat keterangan pelanggan, mengedit, dan menghapus bisa menekan tombol action pada bagian kanan tabel.

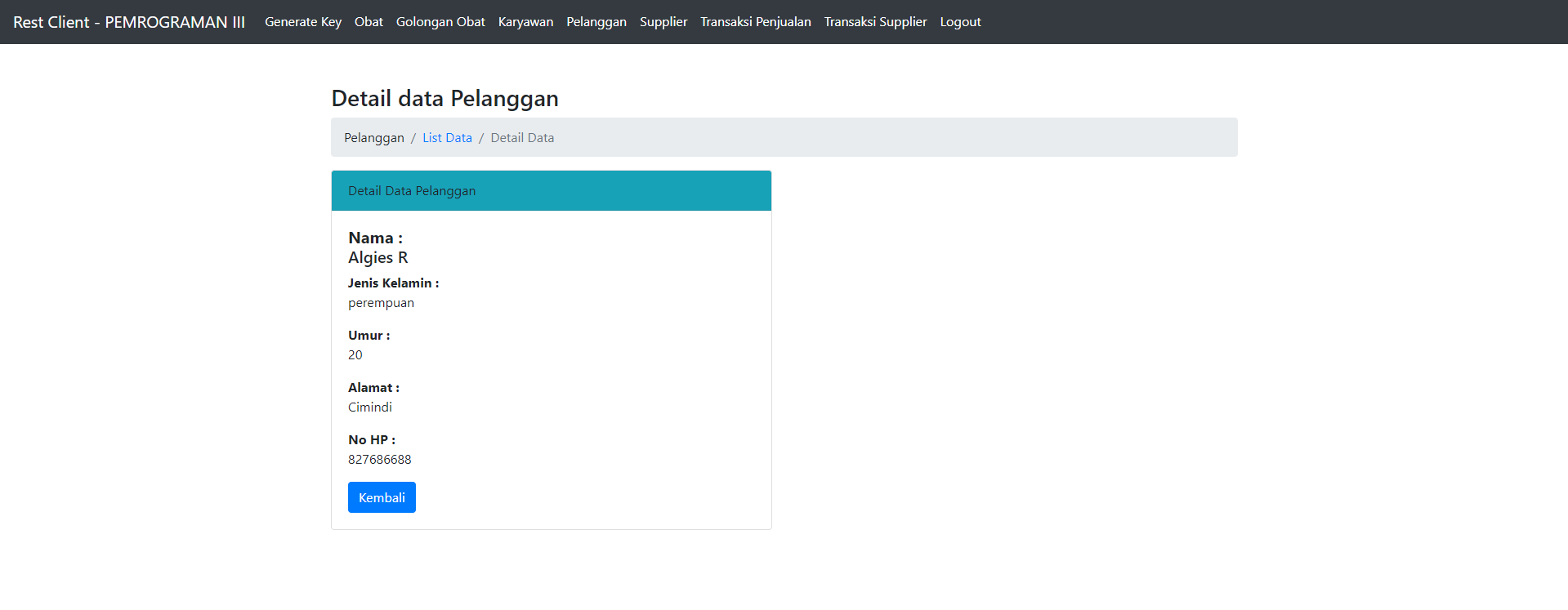
### **View Tambah data pelanggan**





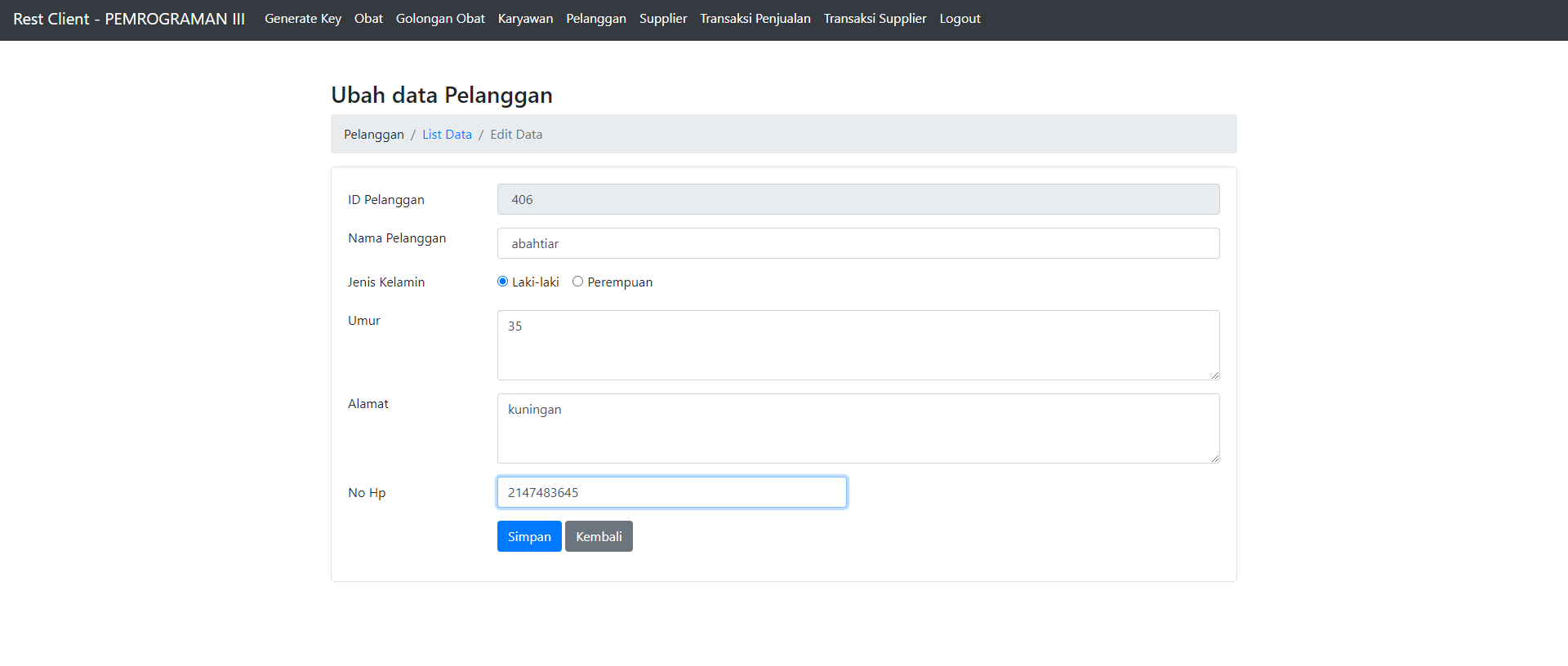
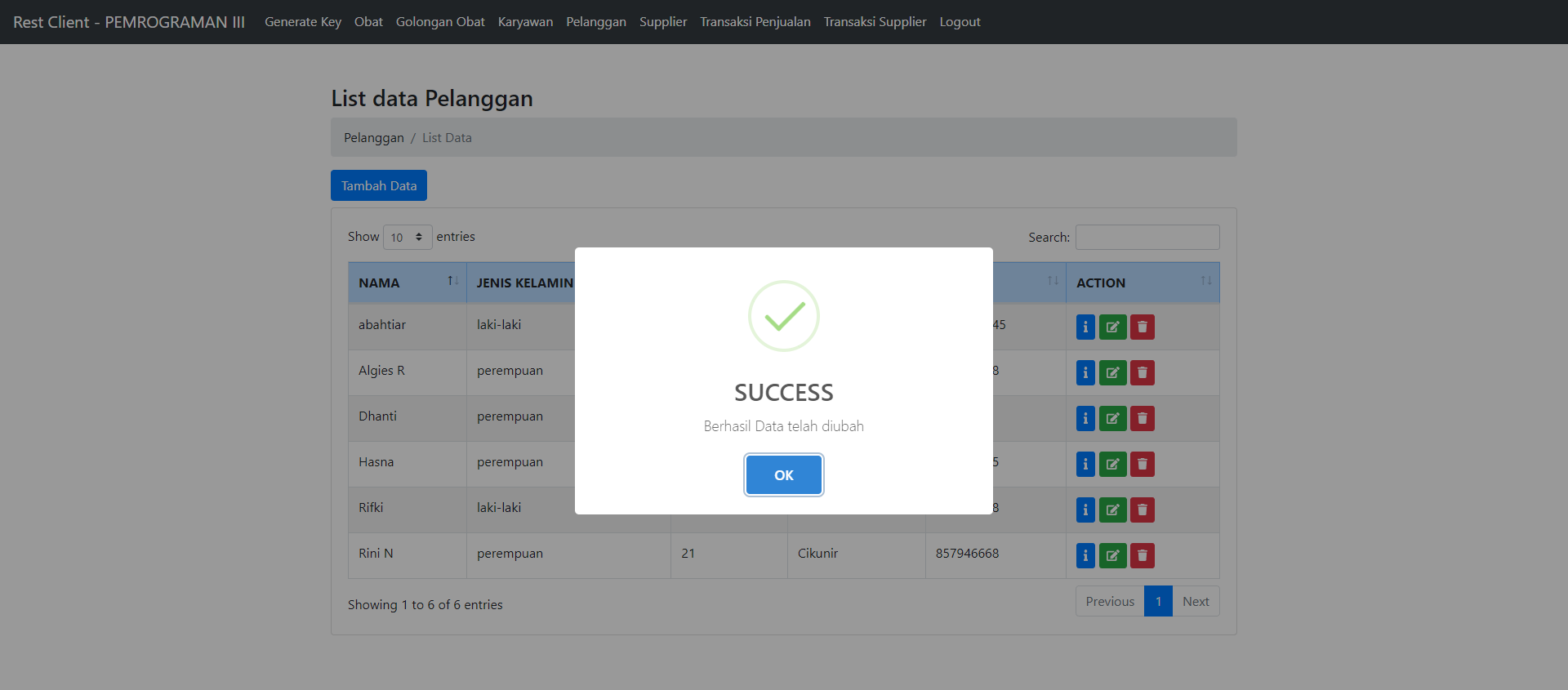
Jika sudah menekan tombol tambah pelanggan maka akan muncul tampilan pelanggan diatas, dapat dilihat untuk menambahkan kita harus memenuhi form dari pelanggan yang ingin ditambahkan seperti id\_pelanggan, nama pelanggan, jenis kelamin, umur, alamat, no hp dan klik simpan untuk menyimpan pelanggan yang ingin ditambahkan, tombol kembali untuk kembali ke halaman pelanggan jika batal menambahkan pelanggan.

### **View List data pelanggan**



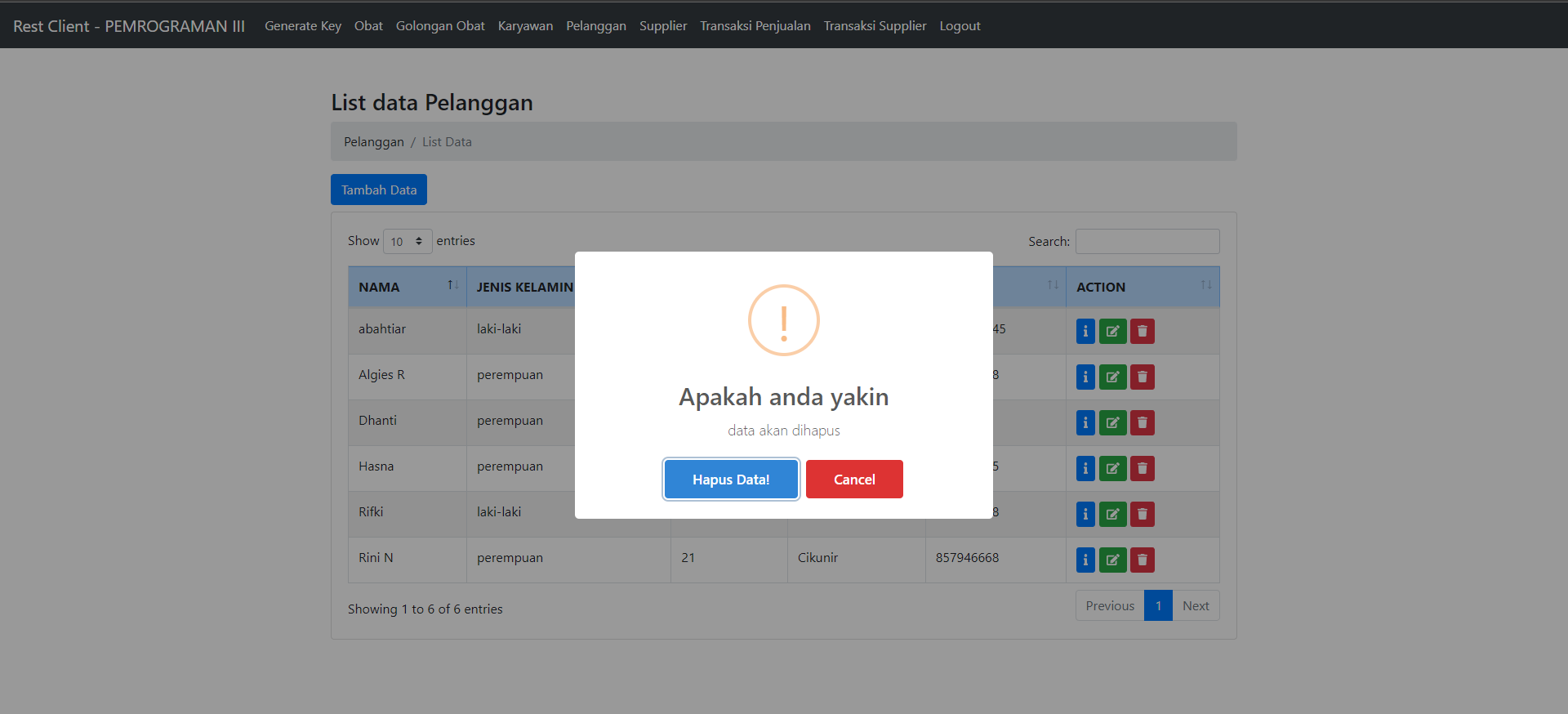
Jika sudah menekan tombol tambah info berwarna biru pada action maka akan muncul tampilan salah satu pelanggan yang dipilih seperti ini.

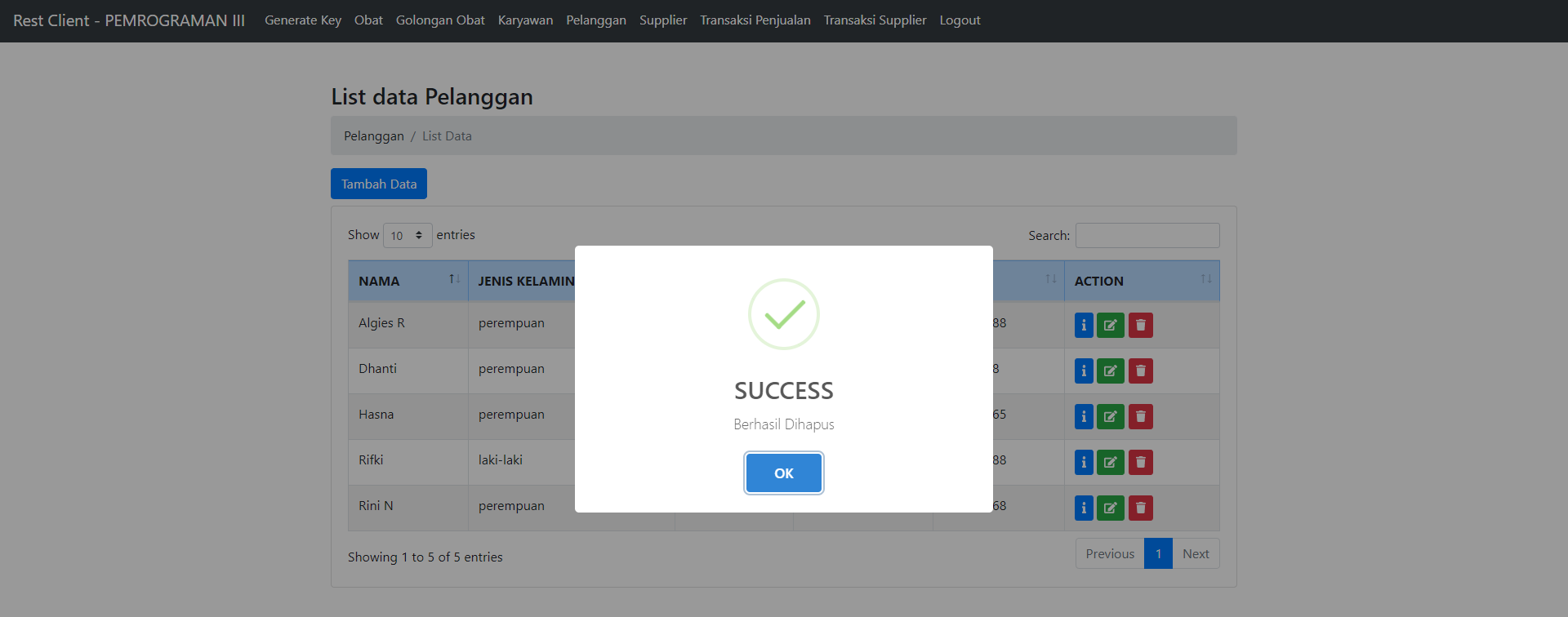
### **View Ubah/Edit data pelanggan**

Jika sudah menekan tombol tambah edit pelanggan berwarna hijau pada action maka akan muncul tampilan edit simpan pelanggan berhasil seperti ini.

### **View Delete/Hapus data Pelanggan**





Jika sudah menekan tombol hapus data pelanggan berwarna merah pada action maka akan muncul tampilan hapus berhasil akan seperti ini.