

**13S3215 | Sistem Paralel Terdistribusi**

**PROYEK SPT**

**DATA MAHASISWA**



**Oleh:**

|                 |                                    |
|-----------------|------------------------------------|
| <b>11S19001</b> | <b>Bungaran Martua Pakpahan</b>    |
| <b>11S19006</b> | <b>Daniel Fernandez Lumbanraja</b> |
| <b>11S19023</b> | <b>Sarah Oktavia br Pasaribu</b>   |

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**

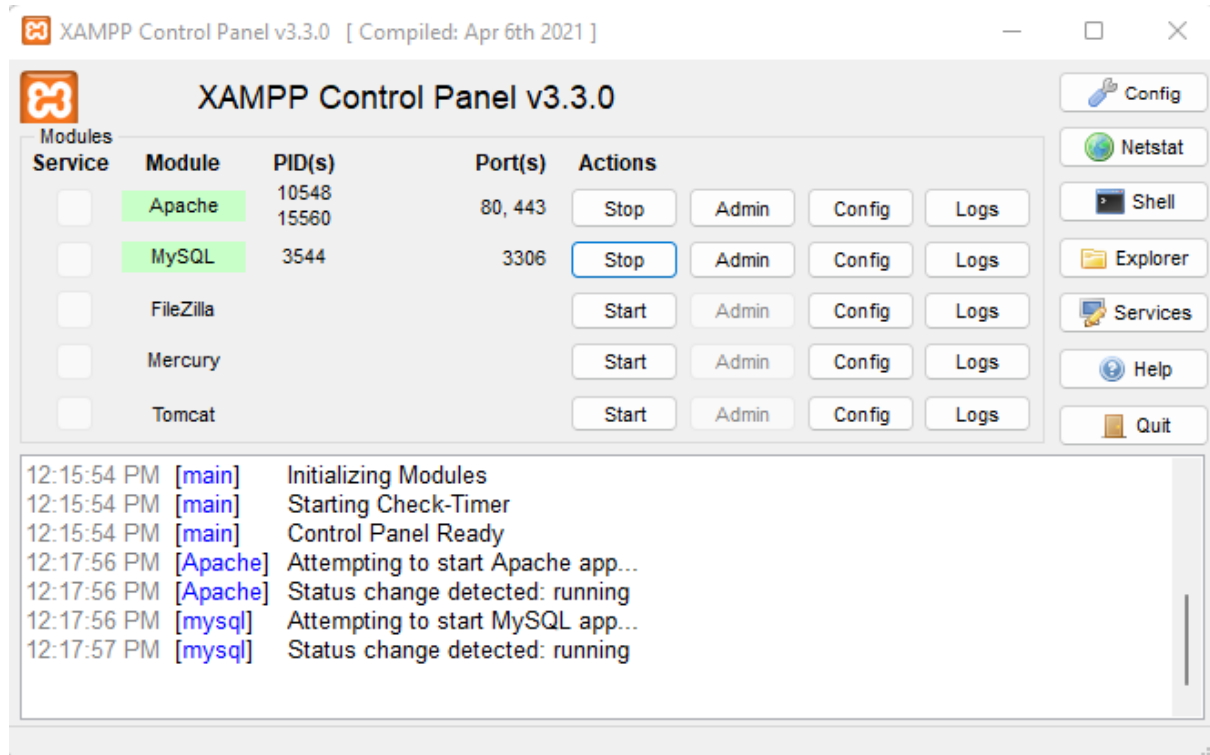
**INSTITUT TEKNOLOGI DEL**

**T.A 2021/2022**

## **BERIKUT ADALAH PANDUAN MENJALANKAN APLIKASI DATA MAHASISWA**

### **Pertama**

Kita buka dulu XAMPP yang berguna untuk sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.



## \*Bagian BackEnd

Pada Back End untuk pembuatan service website kami, kami menggunakan Tech Stack Laravel, adapun resource aplikasi yang diperlukan ialah:

1. Visual Studio Code
2. Laravel, Composer
3. Postman

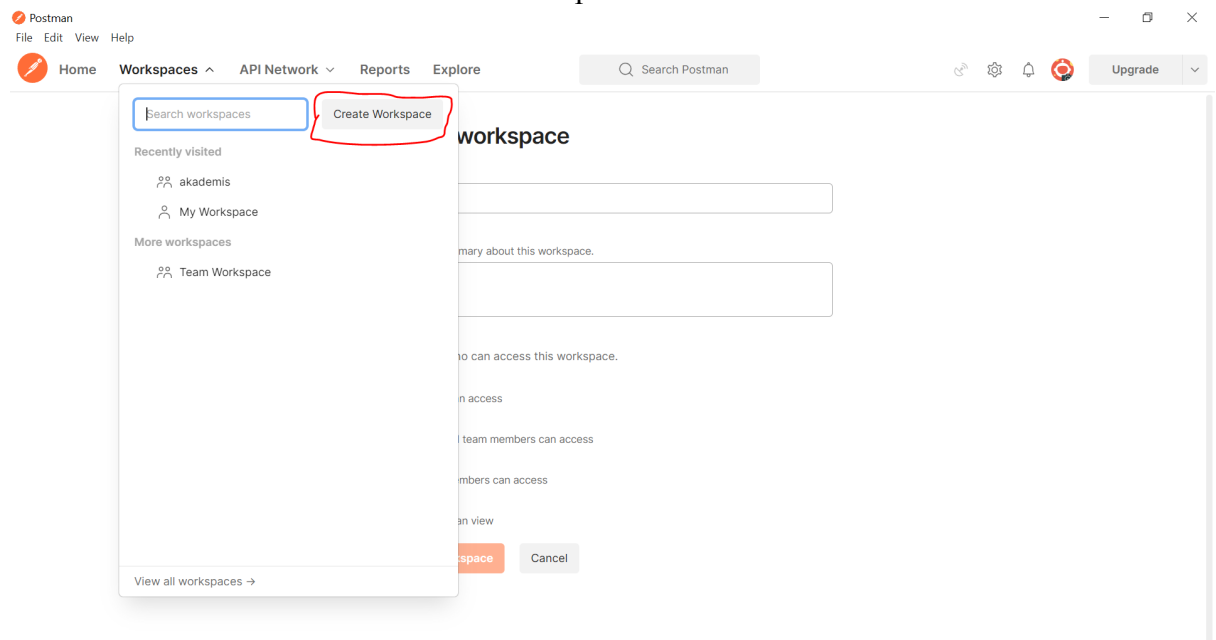
1. Melakukan php artisan serve untuk menjalankan aplikasi laravel

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

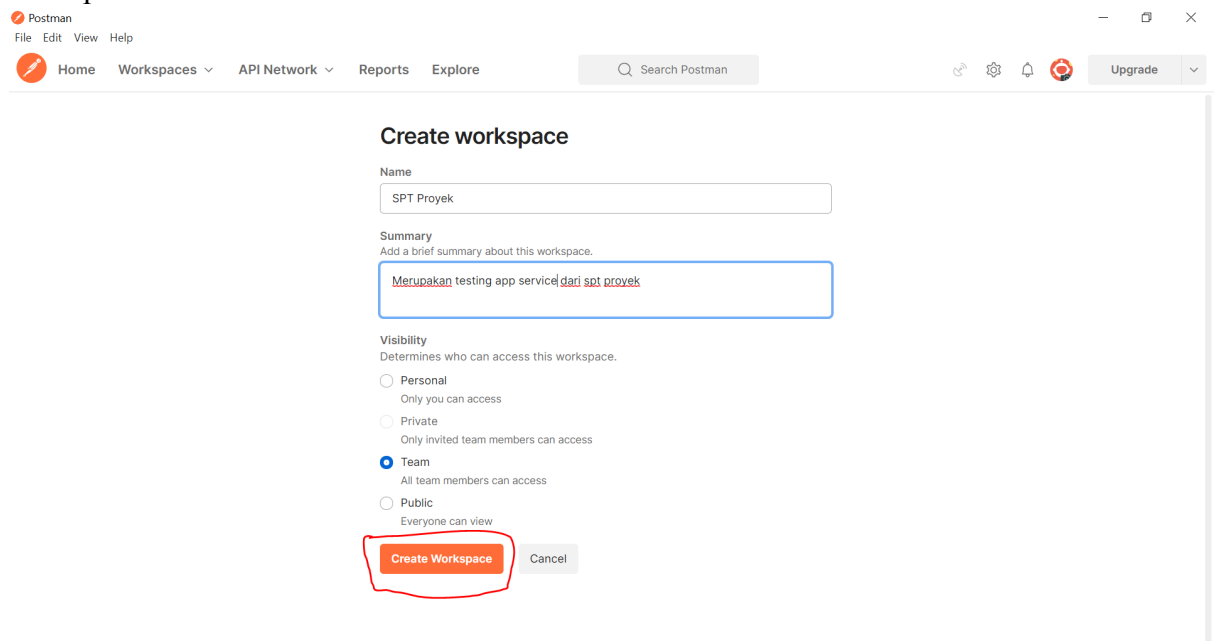
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\SPT Proyek\New\Proyek SPT> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Wed Jun 1 10:12:01 2022] PHP 8.0.11 Development Server (http://127.0.0.1:8000) started
[Wed Jun 1 10:23:27 2022] 127.0.0.1:50148 Accepted
[Wed Jun 1 10:23:29 2022] 127.0.0.1:50148 Closing
[Wed Jun 1 10:23:29 2022] 127.0.0.1:50149 Accepted
[Wed Jun 1 10:23:29 2022] 127.0.0.1:50149 [200]: GET /favicon.ico
[Wed Jun 1 10:23:29 2022] 127.0.0.1:50149 Closing
```

2. Membuka Postman dan create sebuah workspace



3. Mengisi inputan form sesuai dengan yang di inginkan, lalu tekan button create workspace



The screenshot shows the 'Create workspace' form in Postman. The form has three main sections: Name, Summary, and Visibility. The 'Name' field contains 'SPT Proyek'. The 'Summary' field contains 'Merupakan testing app service dari spt proyek'. The 'Visibility' section has four radio buttons: Personal, Private, Team (selected), and Public. The 'Create Workspace' button is highlighted with a red rectangle.

**Create workspace**

**Name**

SPT Proyek

**Summary**

Add a brief summary about this workspace.

Merupakan testing app service dari spt proyek

**Visibility**

Determines who can access this workspace.

☐ Personal  
Only you can access

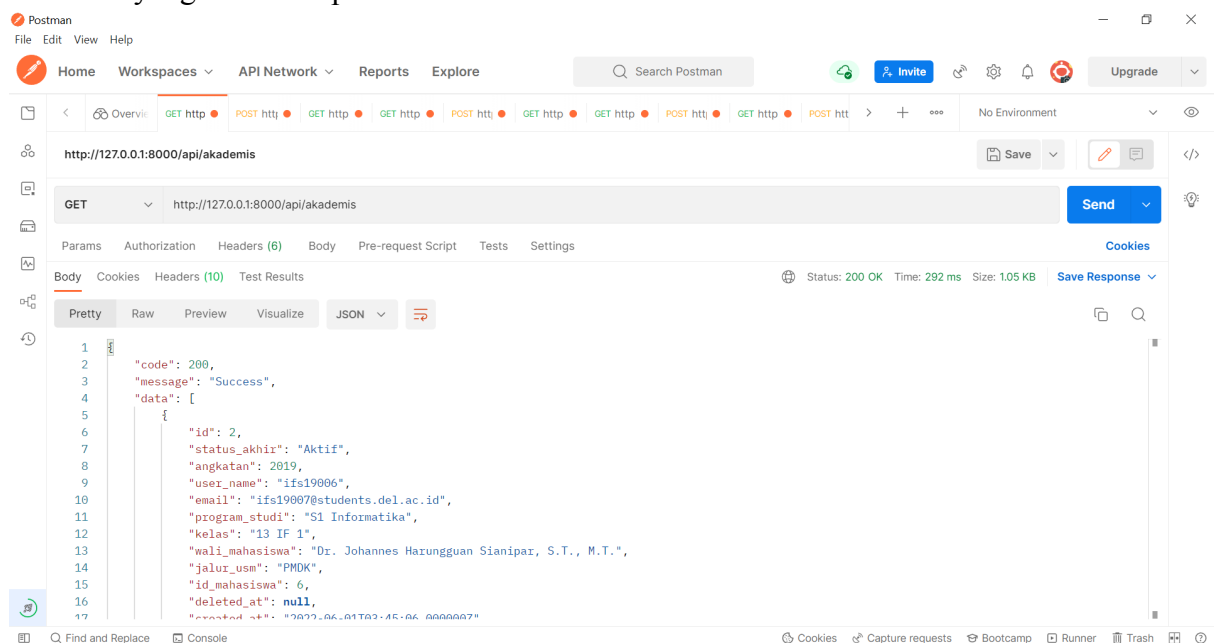
☐ Private  
Only invited team members can access

☒ Team  
All team members can access

☐ Public  
Everyone can view

**Create Workspace** Cancel

4. Melakukan testing dari data akademis, untuk menampilkan seluruh inputan data akademis yang telah di input



The screenshot shows the Postman API client interface. A GET request is sent to the endpoint 'http://127.0.0.1:8000/api/akademis'. The response is a JSON object with a status of 200 OK. The response body is displayed in the 'Body' tab, showing a successful response with a list of academic data.

**GET** http://127.0.0.1:8000/api/akademis

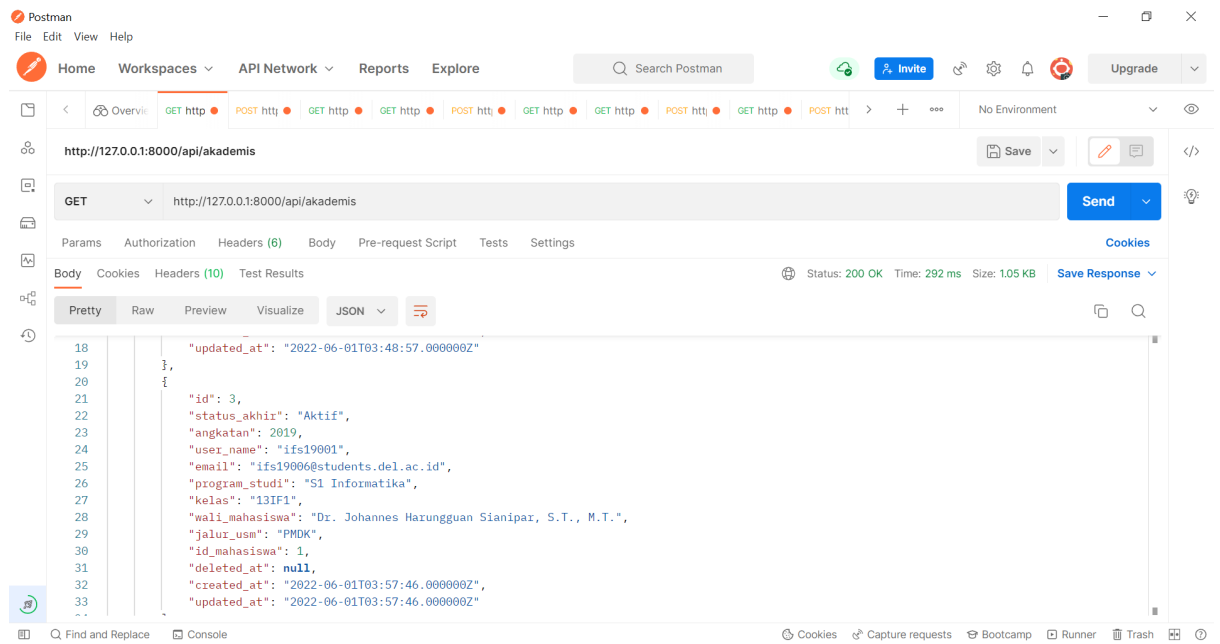
Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

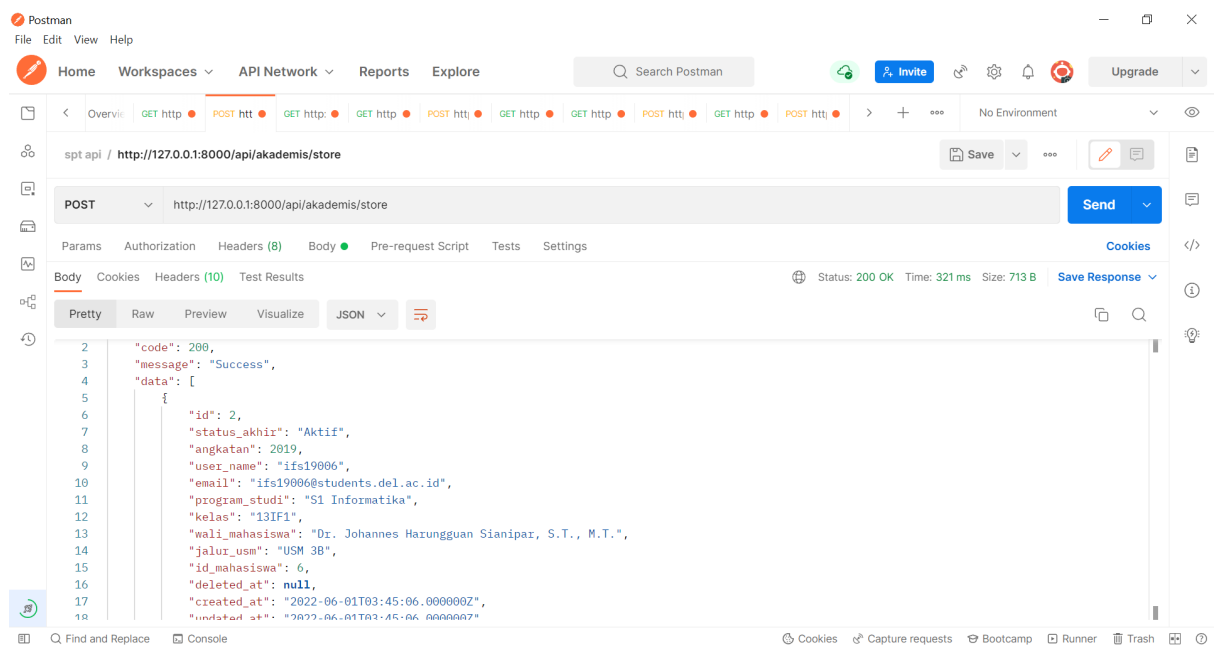
Body Cookies Headers (10) Test Results

Status: 200 OK Time: 292 ms Size: 1.05 KB Save Response

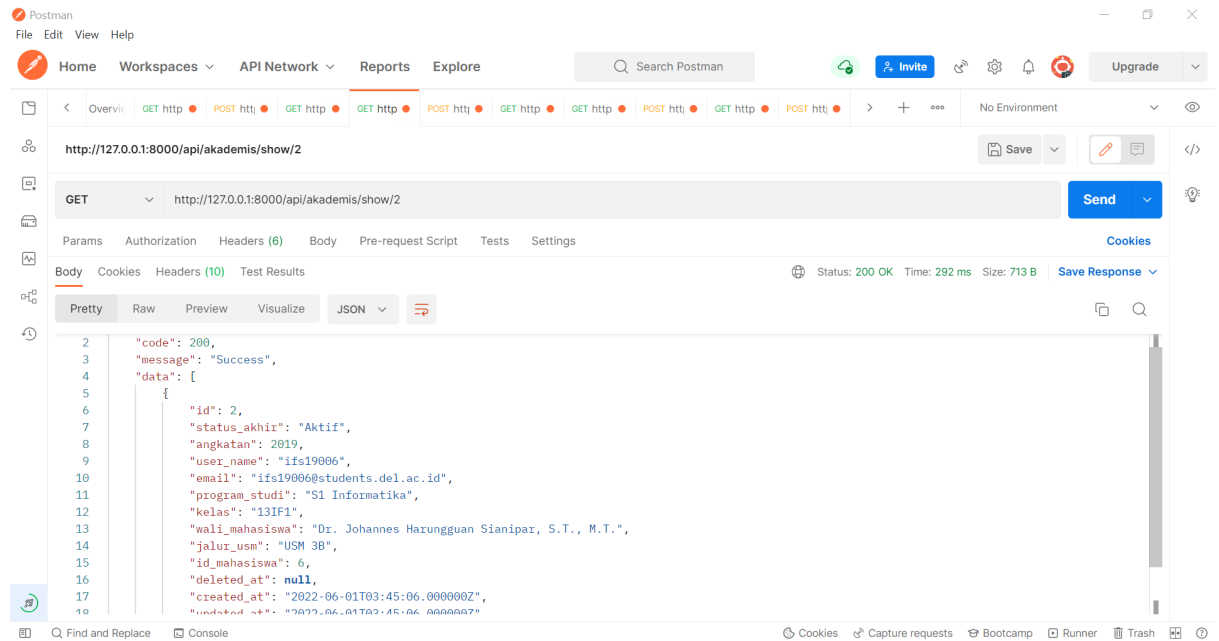
```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": [
5     {
6       "id": 2,
7       "status_akhir": "Aktif",
8       "angkatan": 2019,
9       "user_name": "ifs19006",
10      "email": "ifs19007@students.del.ac.id",
11      "program_studi": "SI Informatika",
12      "kelas": "13 IF 1",
13      "wali_mahasiswa": "Dr. Johannes Hazungguan Sianipar, S.T., M.T.",
14      "jalur_usm": "PMDK",
15      "id_mahasiswa": 6,
16      "deleted_at": null,
17      "created_at": "2022-06-01T02:45:06.000000Z"
```



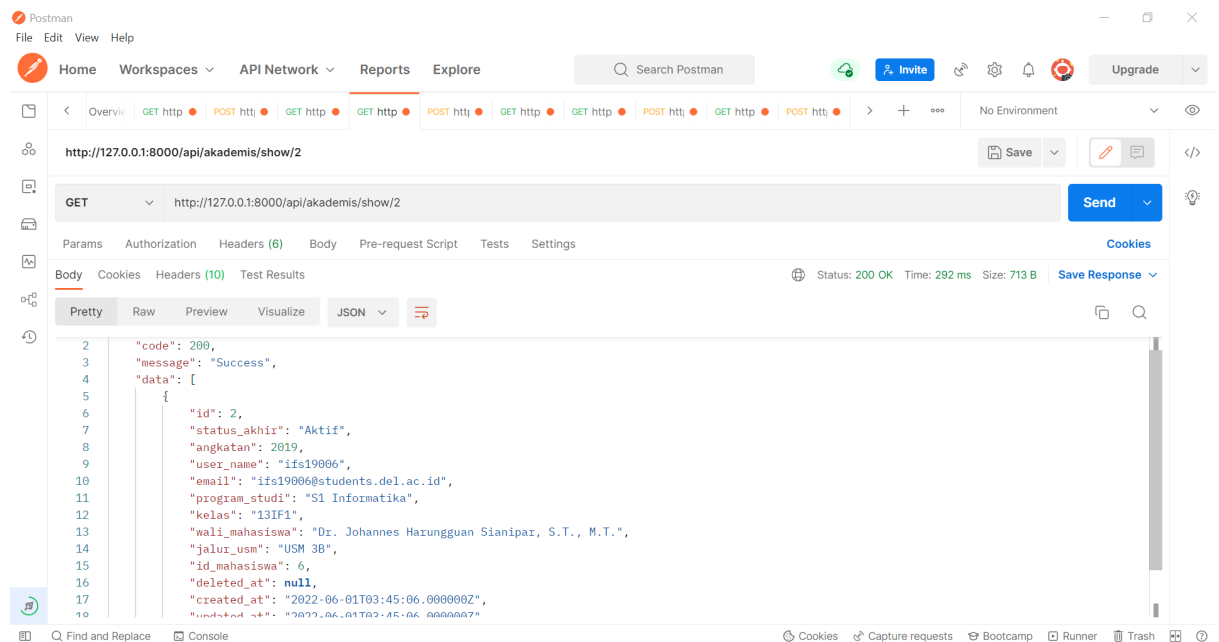
## 5. Melakukan testing dari data akademis, untuk menginput data pada data akademis



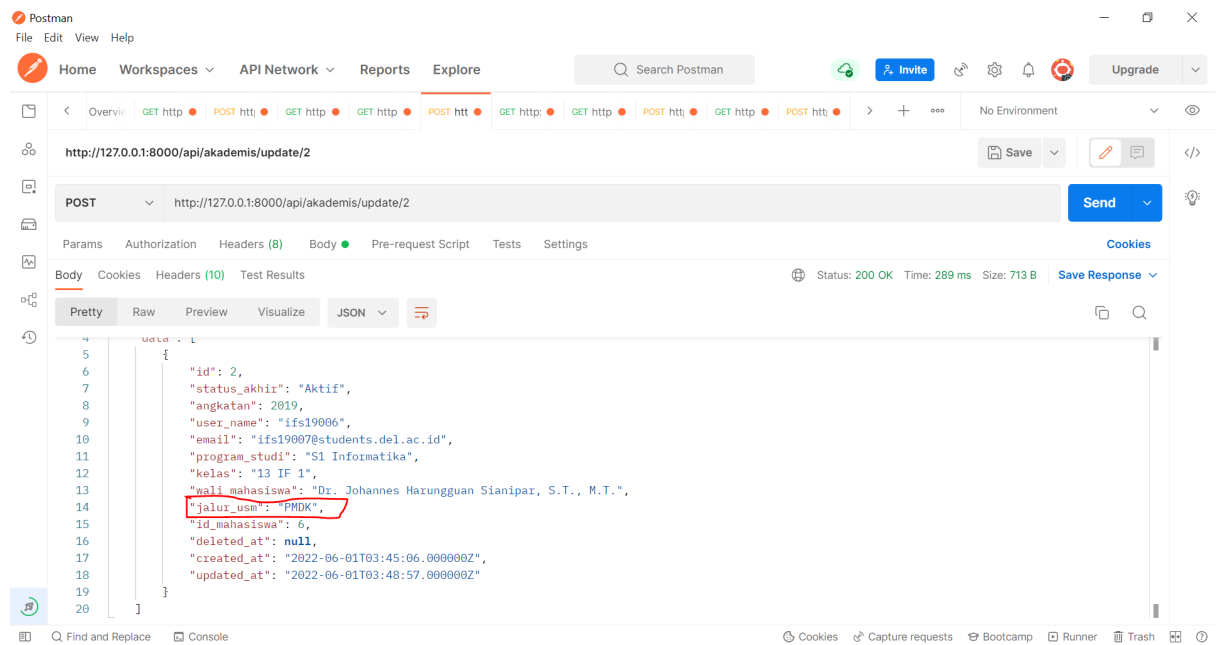
6. Melakukan testing pada data akademis, untuk menampilkan salah satu data yang di input berdasarkan id



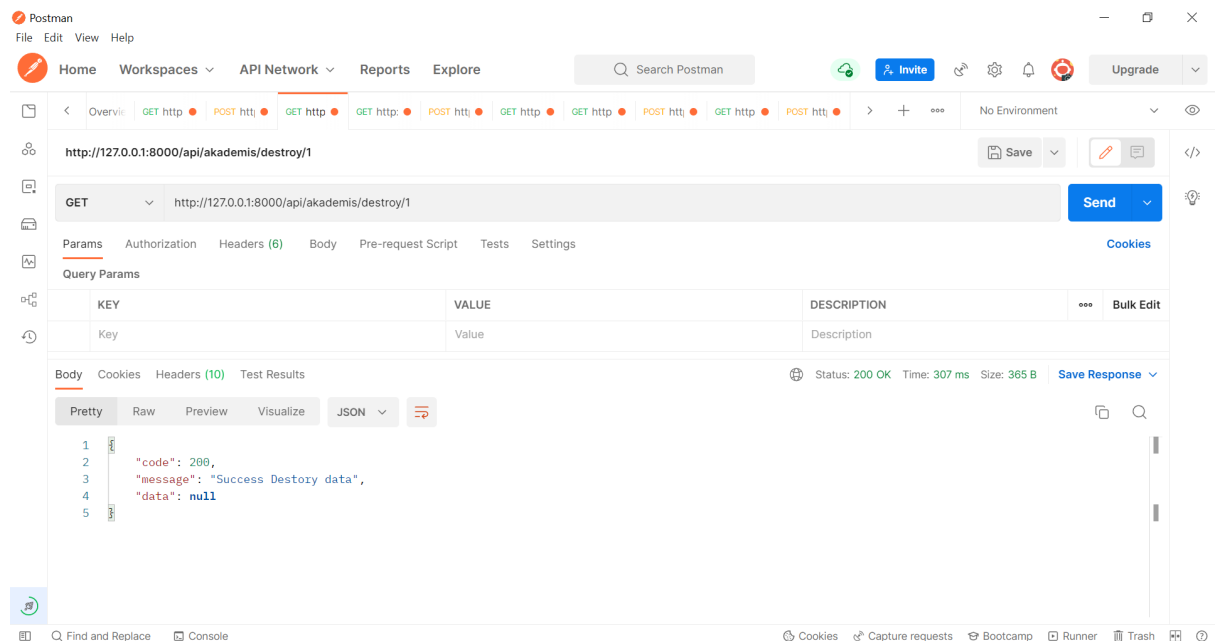
7. Melakukan testing pada data akademis, untuk mengedit salah satu data  
Sebelum edit/update data:



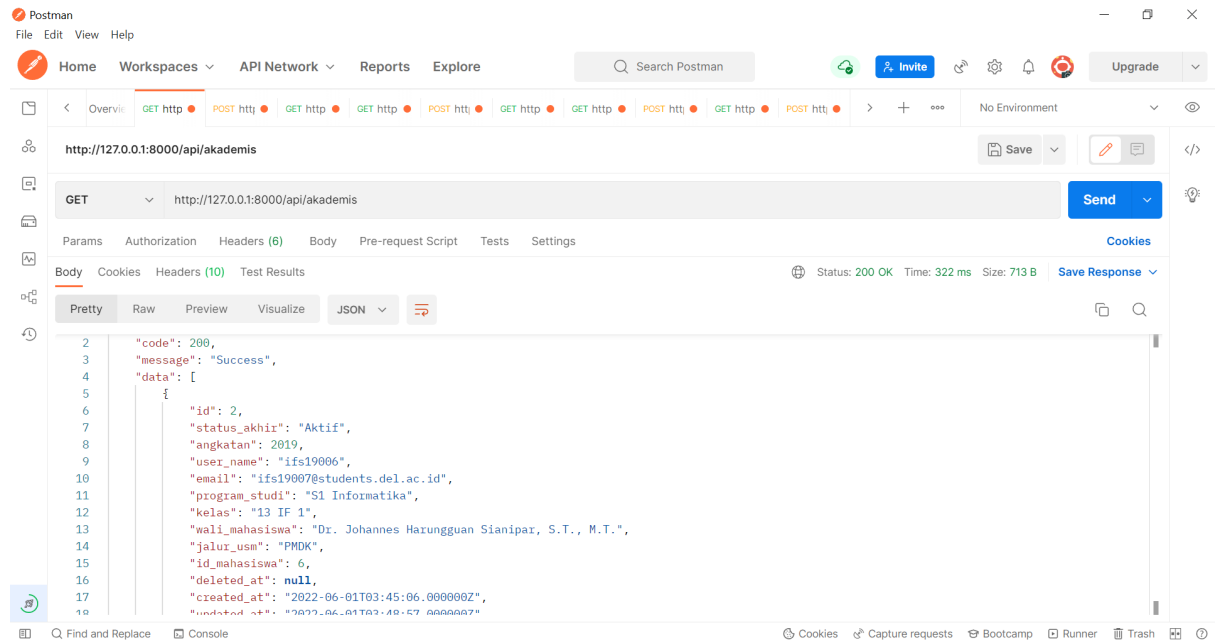
Sesudah edit/update data:



## 8. Melakukan testing pada data akademis, untuk menghapus salah satu data akademis



Ketika menampilkan seluruh data yang masuk apa data akademis, hanya akan menampilkan data pada id 6, seperti gambar berikut:



Untuk data lainnya menggunakan metode yang sama, hanya saja berbeda pada url dimana url dari data-data yang dibuat yaitu sebagai berikut:

- Data Akademis  
<http://127.0.0.1:8000/api/akademis>
- Data Orang Tua  
<http://127.0.0.1:8000/api/orangtua>
- Data Asal Sekolah  
<http://127.0.0.1:8000/api/asalsekolah>
- Data Asrama  
<http://127.0.0.1:8000/api/asrama>

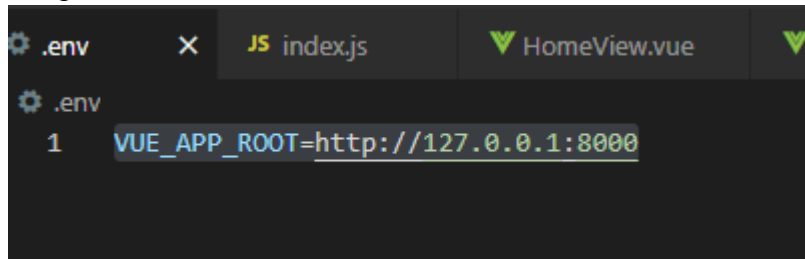


## \*Bagian Front End

Front End ini memakai Vue Js, dimana untuk menghubungkan Vue js nya dari Back End.

**Pertama-tama**, kita membuat **.env** pada Vue js nya ini berguna untuk menghubungkan url yang dari laravel, dan juga untuk mengambil root api nya supaya biar bisa lebih reusable.

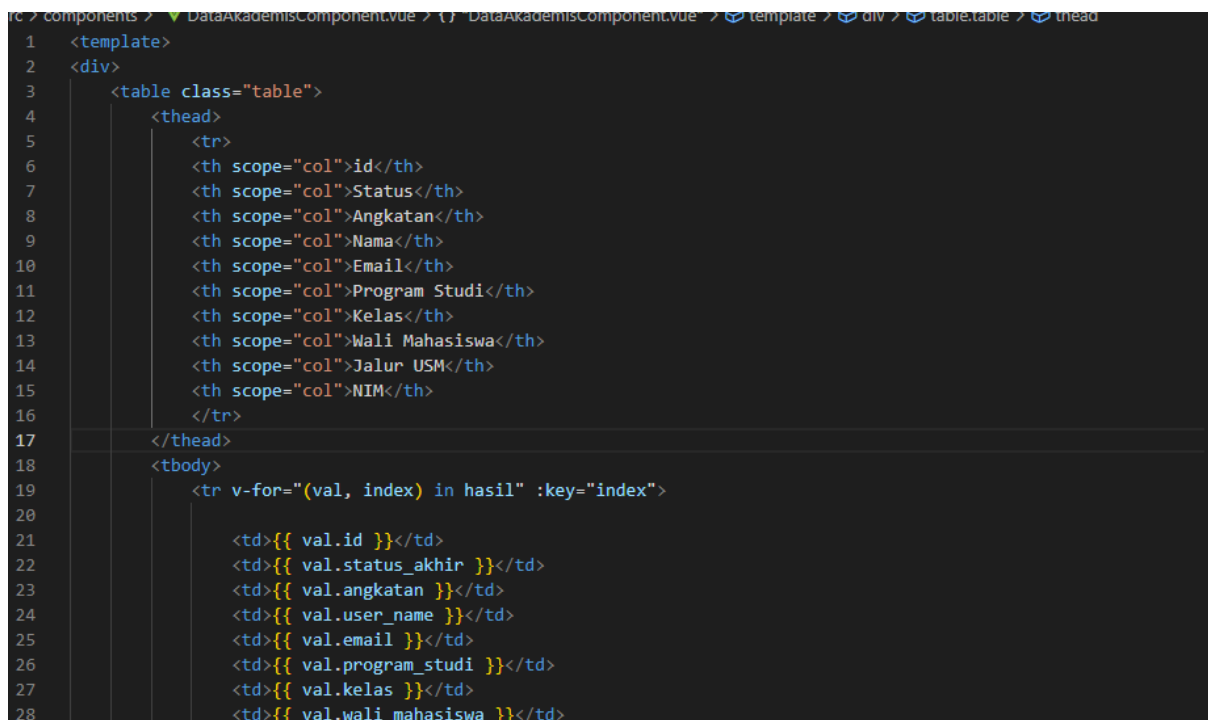
Tampilan kode.env



```
.env
1 VUE_APP_ROOT=http://127.0.0.1:8000
```

**Kedua**, kita membuat Componen nya, dimana component ini berfungsi untuk memungkinkan kita untuk membuat suatu fitur lebih cepat karena kita tidak perlu membuatnya dari awal dan tinggal memanfaatkan component yang sudah ada, selain itu juga menghindari kita dari membuat kode yang sama di component yang lain.

- DataAkademisComponent.vue



```
<template>
<div>
  <table class="table">
    <thead>
      <tr>
        <th scope="col">id</th>
        <th scope="col">Status</th>
        <th scope="col">Angkatan</th>
        <th scope="col">Nama</th>
        <th scope="col">Email</th>
        <th scope="col">Program Studi</th>
        <th scope="col">Kelas</th>
        <th scope="col">Wali Mahasiswa</th>
        <th scope="col">Jalur USM</th>
        <th scope="col">NIM</th>
      </tr>
    </thead>
    <tbody>
      <tr v-for="(val, index) in hasil" :key="index">
        <td>{{ val.id }}</td>
        <td>{{ val.status_akhir }}</td>
        <td>{{ val.angkatan }}</td>
        <td>{{ val.user_name }}</td>
        <td>{{ val.email }}</td>
        <td>{{ val.program_studi }}</td>
        <td>{{ val.kelas }}</td>
        <td>{{ val.wali_mahasiswa }}</td>
      </tr>
    </tbody>
  </table>
</div>
</template>
```

```

src > components > ▼ DataAkademisComponent.vue > {} "DataAkademisComponent.vue" > template > div > ta
30         <td>{{ val.id_mahasiswa}}</td>
31     </tr>
32 </tbody>
33 </table>
34
35
36
37 </div>
38 </template>
39
40 <script>
41 import axios from 'axios'
42
43 export default({
44     data(){
45         return{
46             hasil:[]
47         }
48     },
49     mounted(){
50         this.getData()
51     },
52     methods:{
53         async getData(){
54             axios.get(`http://127.0.0.1:8000/api/akademis`)
55                 .then((res)=> {
56                     this.hasil = res.data.data
57                 })
58             .catch((e)=> {
59                 console.log(e)
60             });
61         },
62     }
63 })
64 </script>
65
66 <style>
67 table{width: 200%; font-size: 12px;}
68
69 </style>
70

```

- DataAsalSekolahComponent.vue

```
1 <template>
2 <div>
3   <table class="table">
4     <thead>
5       <tr>
6         <th scope="col">id</th>
7         <th scope="col">No Ijazah SMA</th>
8         <th scope="col">Nama SMA</th>
9         <th scope="col">Alamat SMA</th>
10        <th scope="col">Kabupaten SMA</th>
11        <th scope="col">Telepon SMA</th>
12        <th scope="col">Kode Pos SMA</th>
13        <th scope="col">Id Mahasiswa</th>
14      </tr>
15    </thead>
16    <tbody>
17      <tr v-for="(val, index) in hasil" :key="index">
18
19        <td>{{ val.id }}</td>
20        <td>{{ val.no_ijazah_sma }}</td>
21        <td>{{ val.nama_sma }}</td>
22        <td>{{ val.alamat_sma }}</td>
23        <td>{{ val.kabupaten_sma }}</td>
24        <td>{{ val.telepon_sma }}</td>
25        <td>{{ val.kode_pos_sma }}</td>
26        <td>{{ val.id_mahasiswa }}</td>
27
28      </tr>
29    </tbody>
30  </table>
31
32
33 </div>
34 </template>
35
36 <script>
37 import axios from 'axios'
38
39 export default({
40   data(){
41     return{
42       hasil:[]
43     }
44   },
45   mounted(){
46     this.getData()
47   },
48   methods:{
49     async getData(){
50       axios.get(`http://127.0.0.1:8000/api/asalsekolah`)
51         .then((res)=> {
52           this.hasil = res.data.data
53         })
54       .catch((e)=> {
```

```

53     })
54     .catch((e)=> {
55         console.log(e)
56     });
57 },
58 }
59
60 })
61 </script>
62 <style>
63 table{width: 200%; font-size: 9px;}
64
65 </style>
66

```

- DataAsramaComponen.vue

```

src > components > DataAsramaComponent.vue > {} "DataAsramaComponent.vue" > template > div > table.table > tbody >
1 <template>
2 <div>
3     <table class="table">
4         <thead>
5             <tr>
6                 <th scope="col">id</th>
7                 <th scope="col">Nama Asrama</th>
8                 <th scope="col">Kamar</th>
9                 <th scope="col">Pengurus Asrama</th>
10                <th scope="col">Nilai Asrama</th>
11                <th scope="col">Id Mahasiswa</th>
12            </tr>
13        </thead>
14        <tbody>
15            <tr v-for="(val, index) in hasil" :key="index">
16                <td>{{ val.id }}</td>
17                <td>{{ val.nama_asrama }}</td>
18                <td>{{ val.kamar }}</td>
19                <td>{{ val.pengurus_asrama }}</td>
20                <td>{{ val.nilai_asrama }}</td>
21                <td>{{ val.id_mahasiswa }}</td>
22            </tr>
23        </tbody>
24    </table>
25
26
27

```

```

28
29 </div>
30 </template>
31
32 <script>
33 import axios from 'axios'
34
35 export default({
36   data(){
37     return{
38       hasil:[]
39     }
40   },
41   mounted(){
42     this.getData()
43   },
44   methods:{
45     async getData(){
46       axios.get(`http://127.0.0.1:8000/api/asrama`)
47         .then((res)=> {
48           this.hasil = res.data.data
49         })
50         .catch((e)=> {
51           console.log(e)
52         });
53     },
54   }
55

```

```

55
56 })
57 </script>
58 <style>
59 table{width: 200%; font-size: 9px;}
60
61 </style>
62

```

- DataOrangtuaComponent.vue

```
1 <template>
2 <div>
3   <table class="table">
4     <thead>
5       <tr>
6         <th scope="col">id</th>
7         <th scope="col">Nama Ayah</th>
8         <th scope="col">Nama Ibu</th>
9         <th scope="col">Alamat Orangtua</th>
10        <th scope="col">Pekerjaan ayah</th>
11        <th scope="col">Penghasilan Ayah</th>
12        <th scope="col">No.Hp Ayah</th>
13        <th scope="col">Pekerjaan Ibu</th>
14        <th scope="col">Penghasilan Ibu</th>
15        <th scope="col">No.Hp Ibu</th>
16        <th scope="col">Jumlah tanggungan</th>
17        <th scope="col">Id Mahasiswa</th>
18      </tr>
19    </thead>
20    <tbody>
21      <tr v-for="(val, index) in hasil" :key="index">
22        <td>{{ val.id }}</td>
23        <td>{{ val.nama_ayah }}</td>
24        <td>{{ val.nama_ibu }}</td>
25        <td>{{ val.alamat_orangtua }}</td>
26        <td>{{ val.pekerjaan_ayah }}</td>
27        <td>{{ val.penghasilan_ayah }}</td>
28        <td>{{ val.pekerjaan_ibu }}</td>
29        <td>{{ val.penghasilan_ibu }}</td>
30        <td>{{ val.no_hp_ibu }}</td>
31        <td>{{ val.jumlah_tanggungan }}</td>
32        <td>{{ val.id_mahasiswa }}</td>
33      </tr>
34    </tbody>
35  </table>
36 </div>
37 </template>
38
39 <script>
40 import axios from 'axios'
41
42 export default({
43   data(){
44     return{
45       hasil:[]
46     }
47   },
48   mounted(){
49     this.getData()
50   },
51   methods:{
52     async getData(){
```

```

54     this.getData()
55   },
56   methods: {
57     async getData() {
58       axios.get(`http://127.0.0.1:8000/api/orangtua`)
59         .then((res) => {
60           this.hasil = res.data.data
61         })
62         .catch((e) => {
63           console.log(e)
64         });
65     },
66   }
67
68 })
69 </script>
70 <style>
71 table{width: 200%; font-size: 9px;}
72
73 </style>
74

```

**Ketiga**, kita panggil url local hostnya untuk menampilkan data yang akan kita pakai.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

vite v2.9.9 dev server running at:

> Local: http://localhost:3000/
> Network: use `--host` to expose

ready in 5125ms.

```

**Keempat**, dipastikan laravel yang sebagai back end nya sudah jalan, supaya **servis** dari API nya terhubung ke vue js nya sebagai front end.

```
PS C:\xampp\htdocs\SPT_Projek_Kelompok_001_006_023-main> php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sat May 28 13:03:54 2022] PHP 8.0.12 Development Server (http://127.0.0.1:8000) started
```

Berikut adalah tampilan dari aplikasi mikroservis DataMahasiswa yang kami buat:  
Data dibawah ini sudah saling terhubung.

### 1. Tampilan DataAkademis

DATA MAHASISWA

Data Akademis

Data Orang Tua

Data Sekolah

Data Asrama

Data Akademis

| id | Status | Angkatan | Nama              | Email                      | Program Studi  | Kelas   | Wali Mahasiswa | Jalur USM | NIM |
|----|--------|----------|-------------------|----------------------------|----------------|---------|----------------|-----------|-----|
| 1  | Aktif  | 2019     | Bungaran Pakpahan | bungaranmart2001@gmail.com | S1 Informatika | 13 IF 1 | JHS            | PMDK      | 1   |

### 2. Tampilan DataOrangtua

DATA MAHASISWA

Data Akademis

Data Orang Tua

Data Sekolah

Data Asrama

Data Orang Tua

| id | Nama Ayah  | Nama Ibu    | Alamat Orangtua | Pekerjaan ayah | Penghasilan Ayah | No.Hp Ayah  | Pekerjaan Ibu | Penghasilan Ibu | No.Hp Ibu   | Jumlah tanggungan | Id Mahasiswa |
|----|------------|-------------|-----------------|----------------|------------------|-------------|---------------|-----------------|-------------|-------------------|--------------|
| 1  | R.Pakpahan | R.Silitonga | Pasar Sipahutar | Wiraswasta     | 1 juta           | 81234345566 | Wiraswasta    | 1 juta          | 81234345566 | 2                 | 0            |



3. Tampilan DataSekolah

DATA MAHASISWA

Data Akademis

Data Orang Tua

Data Sekolah

Data Asrama

| Data Sekolah |               |                |            |               |             |              |              |
|--------------|---------------|----------------|------------|---------------|-------------|--------------|--------------|
| id           | No Ijazah SMA | Nama SMA       | Alamat SMA | Kabupaten SMA | Telepon SMA | Kode Pos SMA | Id Mahasiswa |
| 1            | 12345678      | SMA N.1 Balige | Hinalang   | Toba          | 12345678    | 13211        | 1            |

4. Tampilan Asrama

DATA MAHASISWA

Data Akademis

Data Orang Tua

Data Sekolah

Data Asrama

| Data Asrama |             |       |                 |              |              |
|-------------|-------------|-------|-----------------|--------------|--------------|
| id          | Nama Asrama | Kamar | Pengurus Asrama | Nilai Asrama | Id Mahasiswa |
| 1           | Pniel       | 1     | Pdt.Bungaran    | A            | 1            |