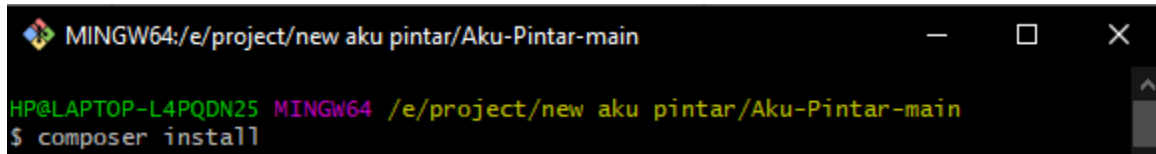
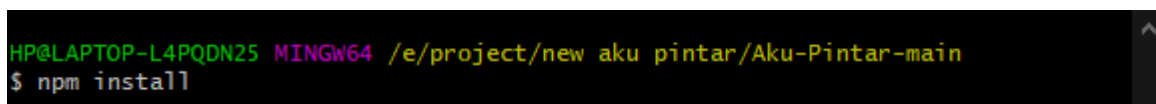


1. Import database akupintar.sql ke mysql
2. Jalankan git bash
3. Ketik composer install



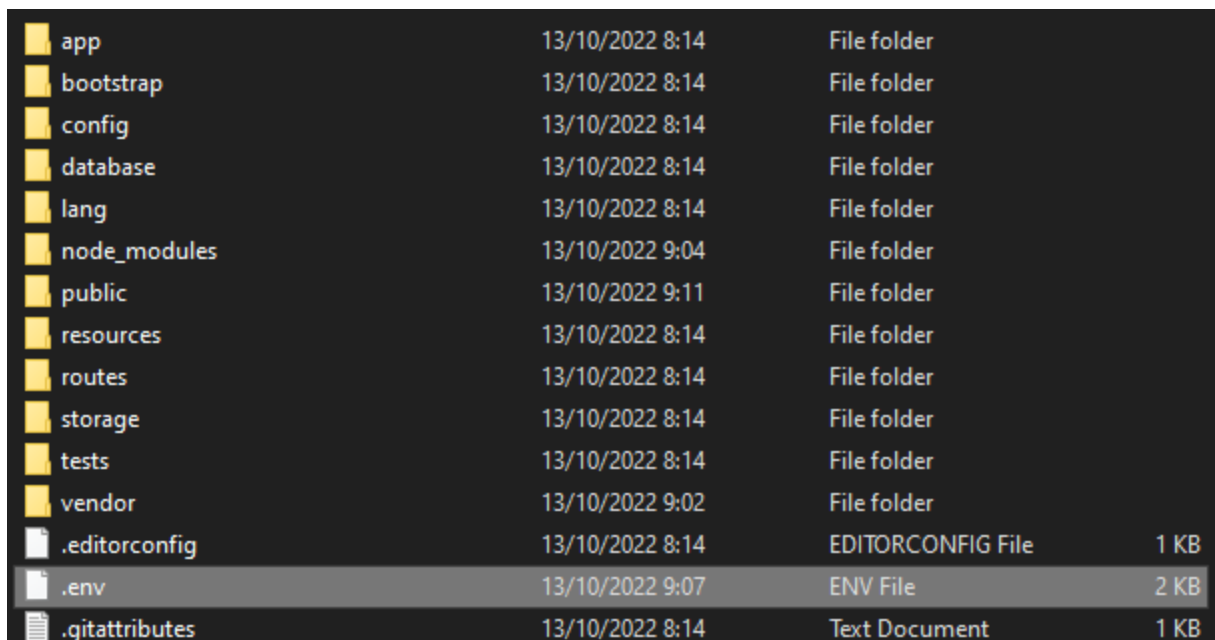
```
MINGW64:/e/project/new aku pintar/Aku-Pintar-main
HP@LAPTOP-L4PQDN25 MINGW64 /e/project/new aku pintar/Aku-Pintar-main
$ composer install
```

4. Ketik npm install



```
HP@LAPTOP-L4PQDN25 MINGW64 /e/project/new aku pintar/Aku-Pintar-main
$ npm install
```

5. Ubah .env.example jadi .env



app	13/10/2022 8:14	File folder	
bootstrap	13/10/2022 8:14	File folder	
config	13/10/2022 8:14	File folder	
database	13/10/2022 8:14	File folder	
lang	13/10/2022 8:14	File folder	
node_modules	13/10/2022 9:04	File folder	
public	13/10/2022 9:11	File folder	
resources	13/10/2022 8:14	File folder	
routes	13/10/2022 8:14	File folder	
storage	13/10/2022 8:14	File folder	
tests	13/10/2022 8:14	File folder	
vendor	13/10/2022 9:02	File folder	
.editorconfig	13/10/2022 8:14	EDITORCONFIG File	1 KB
.env	13/10/2022 9:07	ENV File	2 KB
.gitattributes	13/10/2022 8:14	Text Document	1 KB

6. Buka .env masukan DB_DATABASE=akupintar

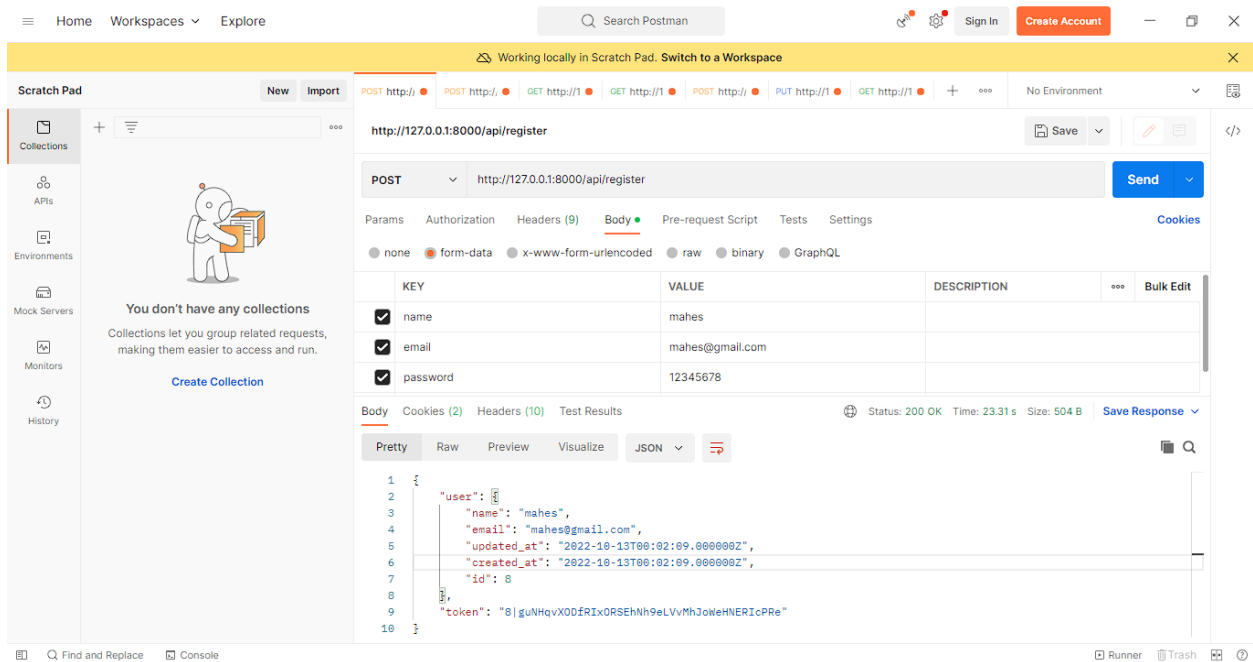
```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=akupintar
DB_USERNAME=root
DB_PASSWORD=
```

7. Ketik php artisan serve

```
HP@LAPTOP-L4PQDN25 MINGW64 /e/project/new aku pintar/Aku-Pintar-main
$ php artisan serve
```

8. Jalankan Postman untuk menguji API

9. API Registrasi User



Membuat data baru, dan mengirim data user ke database users, dan personal_access_token.

10. API Login User

The screenshot shows the Postman interface with a POST request to `http://127.0.0.1:8000/api/login`. The request body is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
email	mahes@gmail.com	
password	12345678	

The response is a JSON object with the following data:

```
1 {
2   "user": {
3     "id": 8,
4     "name": "mahes",
5     "email": "mahes@gmail.com",
6     "email_verified_at": null,
7     "created_at": "2022-10-13T08:02:09.000000Z",
8     "updated_at": "2022-10-13T08:02:09.000000Z"
9   },
10  "token": "9|dhkZ8rUcc4fUuNC55tMf9rDRE6L8J1FQy0LhPLtu"
```

The status is 200 OK, Time: 787 ms, Size: 529 B.

11. API Daftar Kampus

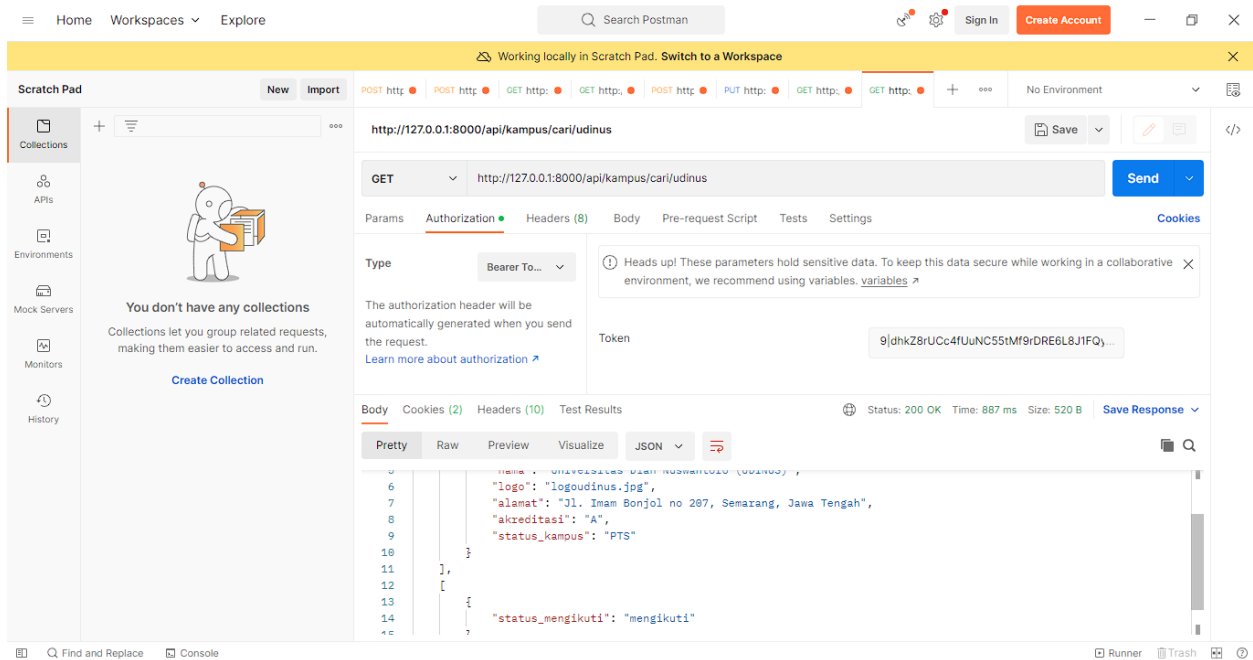
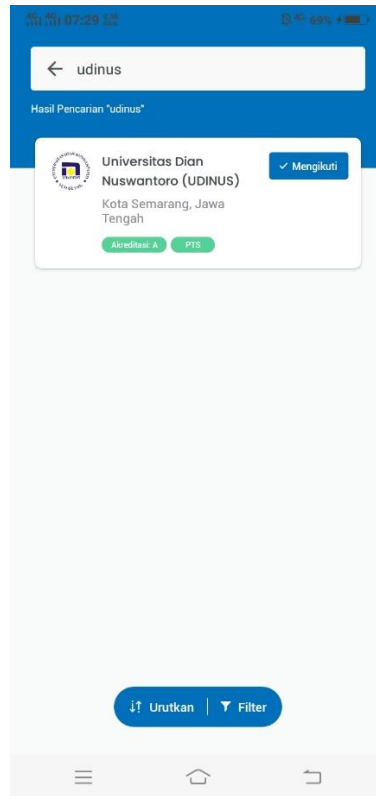
The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/api/kampus`. The request is authorized with a Bearer token. The response is a JSON object with the following data:

```
1 {
2   "id": 1,
3   "nama": "Universitas Dian Nuswantoro (UDINUS)",
4   "logo": "logoudinus.jpg",
5   "akreditasi": "A",
6   "status_kampus": "PTS",
7   "jenis": "SWASTA",
8   "no_telepon": "024-3517261",
9   "fax": "024-3569604",
10 }
```

The status is 200 OK, Time: 773 ms, Size: 1.45 KB.

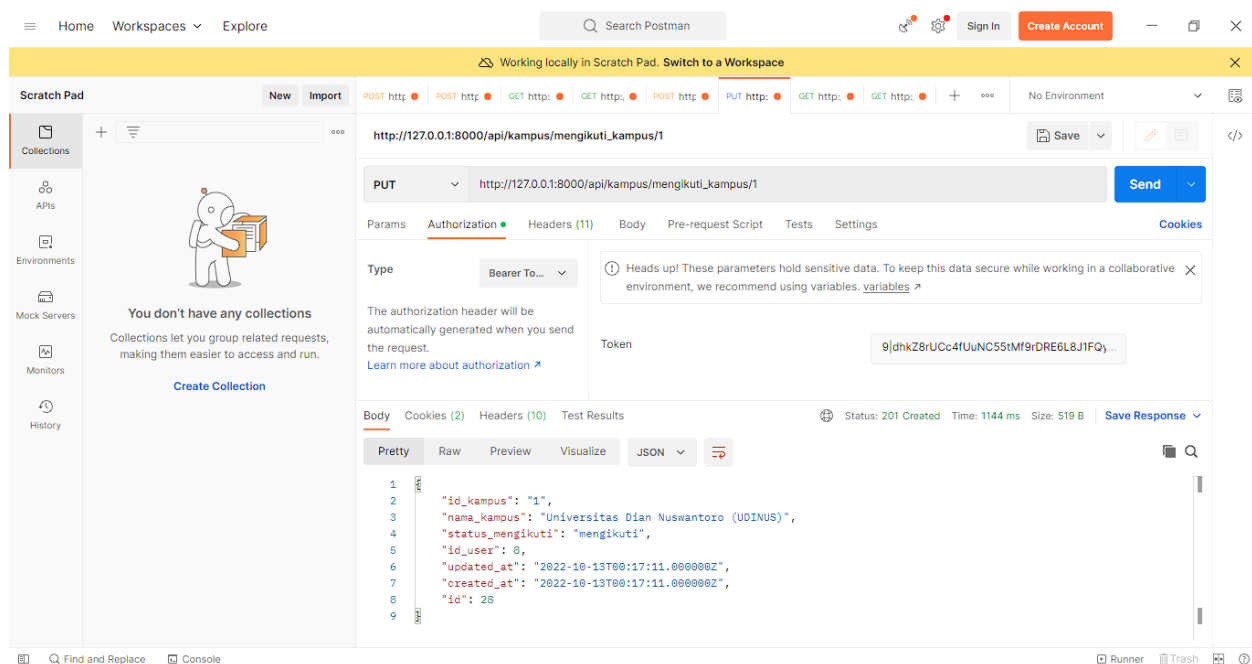
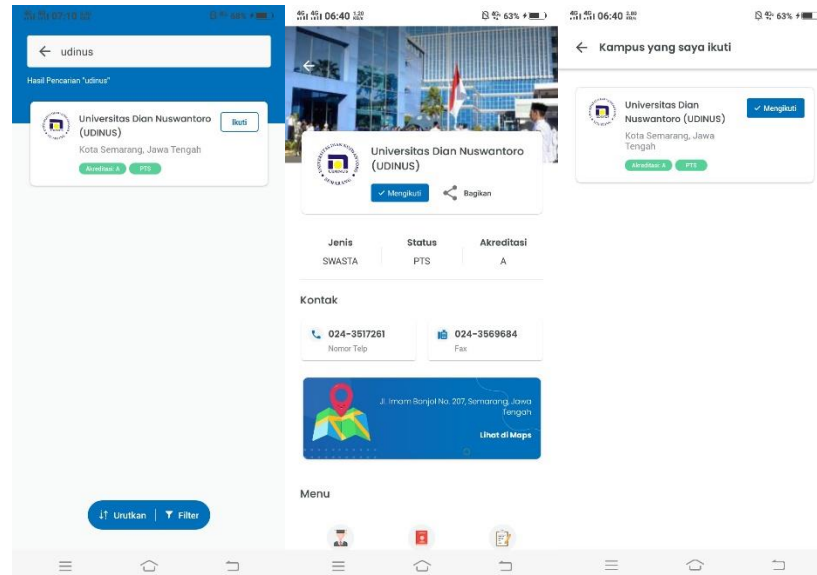
Melihat semua daftar kampus yang ada di database “kampus” dengan cara menggunakan authorization type bearer token dan memasukan token user (mahes) login sebelumnya.

12. API Pencarian Kampus



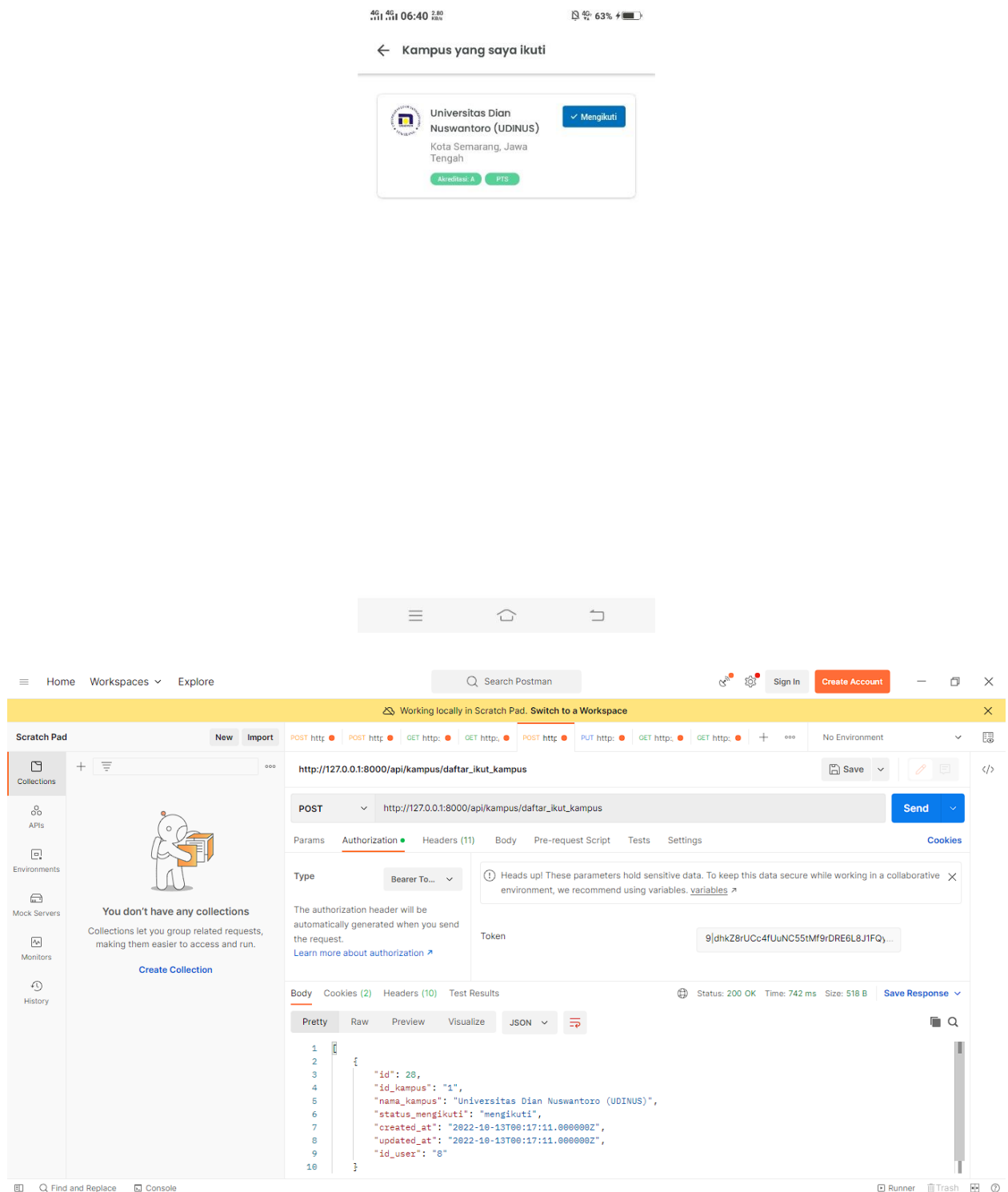
Menampilkan kampus yang dicari (udinus) oleh user dengan url `http://127.0.0.1:8000/api/kampus/cari/udinus` , dan authorization bearer token user (mahes).

13. API Mengikuti Kampus



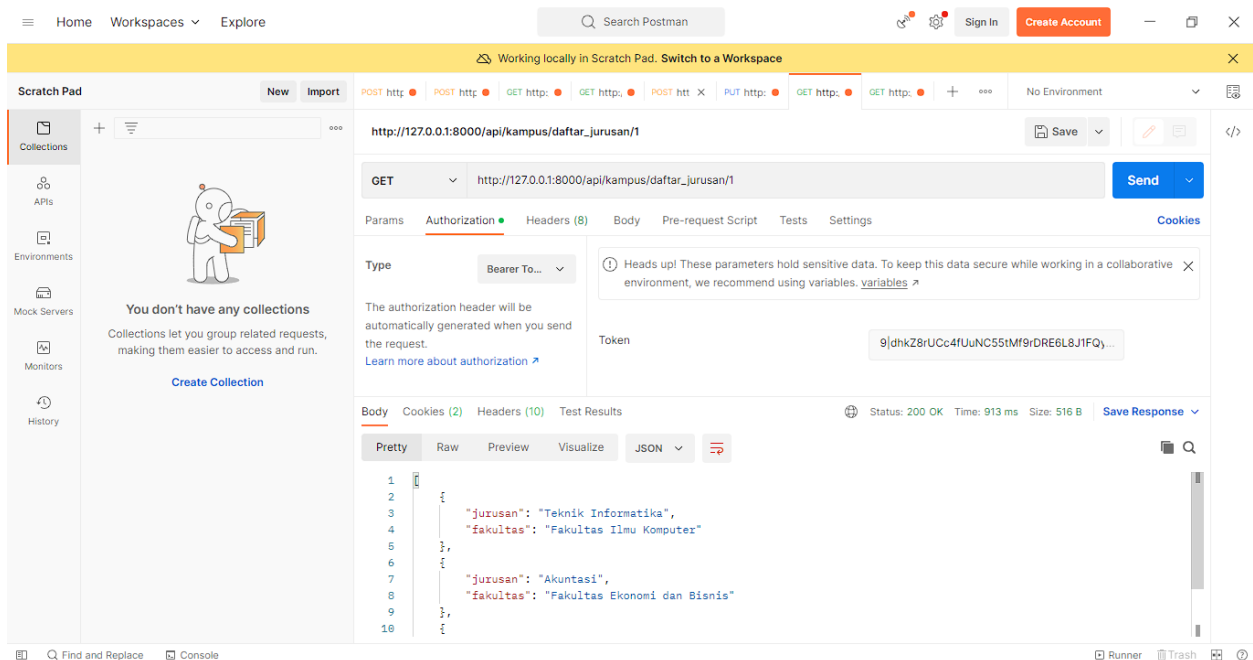
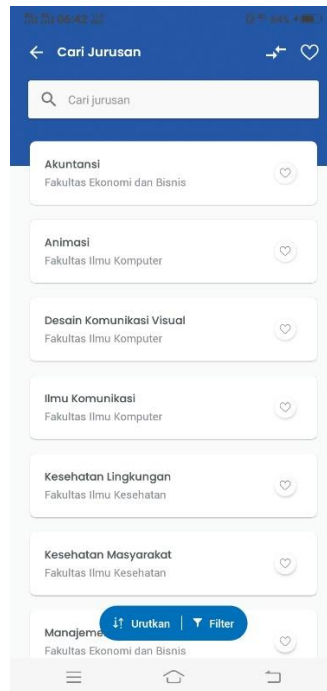
Jika user (Mahes) melakukan mengikuti kampus Universitas Dian Nuswantoro UDINUS (id=1) maka url http://127.0.0.1:8000/api/kampus/mengikuti_kampus/1 dengan authorization type bearer token user mahes. Pada database “mengikuti” menambahkan status_mengikuti user menjadi “mengikuti” jika sebelumnya status_mengikuti=null atau kosong, dan mengganti status_mengikuti jika sebelumnya status_mengikuti=“belum”.

14. API Kampus Yang Saya Ikuti



Menampilkan daftar kampus yang diikuti oleh user (mahes) dengan authorization type bearer token user (mahes) berdasarkan database “mengikuti”.

15. API Daftar Jurusan Kampus



Menampilkan daftar jurusan kampus Universitas Dian Nuswantoro(id=1) dengan url http://127.0.0.1:8000/api/kampus/daftar_jurusan/1.