DigitalSkola

PROJECT 5

BY : DREAM (KELOMPOK 1)

Data Realm Engineers And Maestros

Anggota Kelompok

01

Afroh Fauziah

02

Althaf Nawadir Taqiyyah

03

Andi Rosilala

04

Andrew Bintang Pratama

05

Andrew Fortino Mahardika Suadnya

TUJUAN PROJEK

Project 5 - Create CRUD API Using Flask and Docker

Tools yang harus disiapkan:

- docker
- postman
- visual code

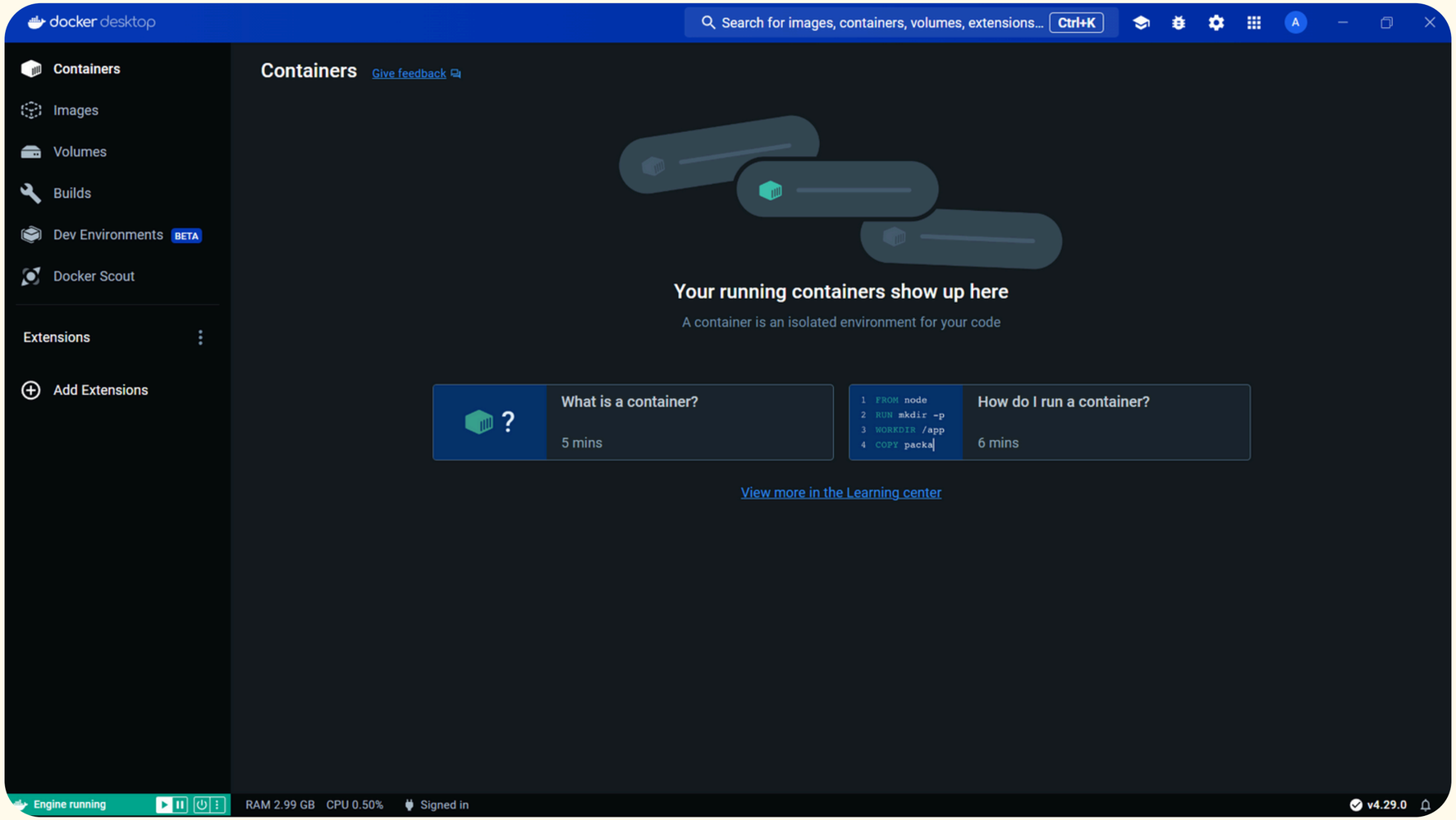
API yang dibuat:

- create user
- update user
- get user
- delete user

Projek ini bertujuan untuk membuat sebuah API CRUD menggunakan Flask dan Docker. Alat yang dibutuhkan adalah Docker untuk manajemen kontainer, Postman untuk pengujian API, dan Visual Studio Code sebagai code editor. API yang akan dibuat mencakup operasi dasar CRUD untuk entitas user: create, update, get, dan delete.

DigitalSkola

STEP 1: Menyalakan Docker Menggunakan Docker Desktop



DigitalSkola

STEP 2: Membuat file docker-compose.yml

File docker-compose.yml ini digunakan untuk mendefinisikan layanan yang akan dijalankan dalam lingkungan Docker. Tujuannya untuk menyiapkan sebuah layanan PostgreSQL, database yang akan digunakan sebagai basis data untuk aplikasi Flask yang akan dibuat.

docker-compose.yml

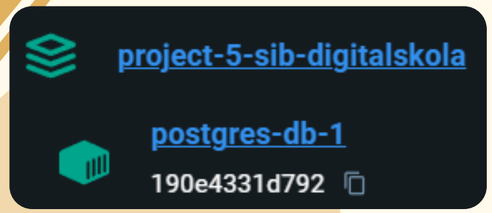
```yaml
version: "2"
services:
  postgres-db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: admin
      POSTGRES_USER: postgres
      PGDATA: /var/lib/postgres/data
      POSTGRES_DB: sample
    volumes:
      - pg-data:/var/lib/postgres/data
    ports:
      - 5432:5432/tcp
volumes:
  pg-data: #docker volume create pg-data
    external: true
```

STEP 3: Menjalankan Kontainer

Command '**docker volume create pg-data**' dijalankan untuk membuat volume Docker dengan nama **pg-data**. Tujuannya untuk menyiapkan tempat penyimpanan data yang akan digunakan oleh kontainer Docker (Postgres).

```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker volume create pg-data
pg-data
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker-compose up -d
[+] Running 15/15
 ✓ postgres-db Pulled                                                                                                    296.3s
   ✓ b0a0cf830b12 Pull complete                                                                                           90.1s
   ✓ dda3d8fbd5ed Pull complete                                                                                            1.8s
   ✓ 283a477db7bb Pull complete                                                                                           11.9s
   ✓ 91d2729fa4d5 Pull complete                                                                                           13.0s
   ✓ 9739ced65621 Pull complete                                                                                           36.9s
   ✓ ae3bb1b347a4 Pull complete                                                                                           22.7s
   ✓ f8406d9c00ea Pull complete                                                                                           24.6s
   ✓ c199bff16b05 Pull complete                                                                                           27.6s
   ✓ e0d55fdb4d15 Pull complete                                                                                          270.8s
   ✓ c1cb13b19080 Pull complete                                                                                           38.8s
   ✓ 873532e5f8c7 Pull complete                                                                                          284.9s
   ✓ 050d9f8c3b1c Pull complete                                                                                           43.2s
   ✓ 710e142705f8 Pull complete                                                                                           45.0s
   ✓ cb628c265f09 Pull complete                                                                                           46.7s
[+] Running 2/2
 ✓ Network project-5-sib-digitalskola_default          Created                                                            0.2s
 ✓ Container project-5-sib-digitalskola-postgres-db-1  Started                                                            3.0s
```

Menjalankan kontainer Docker berdasarkan konfigurasi yang didefinisikan dalam file "docker-compose.yml" dengan command **'docker-compose up -d'**



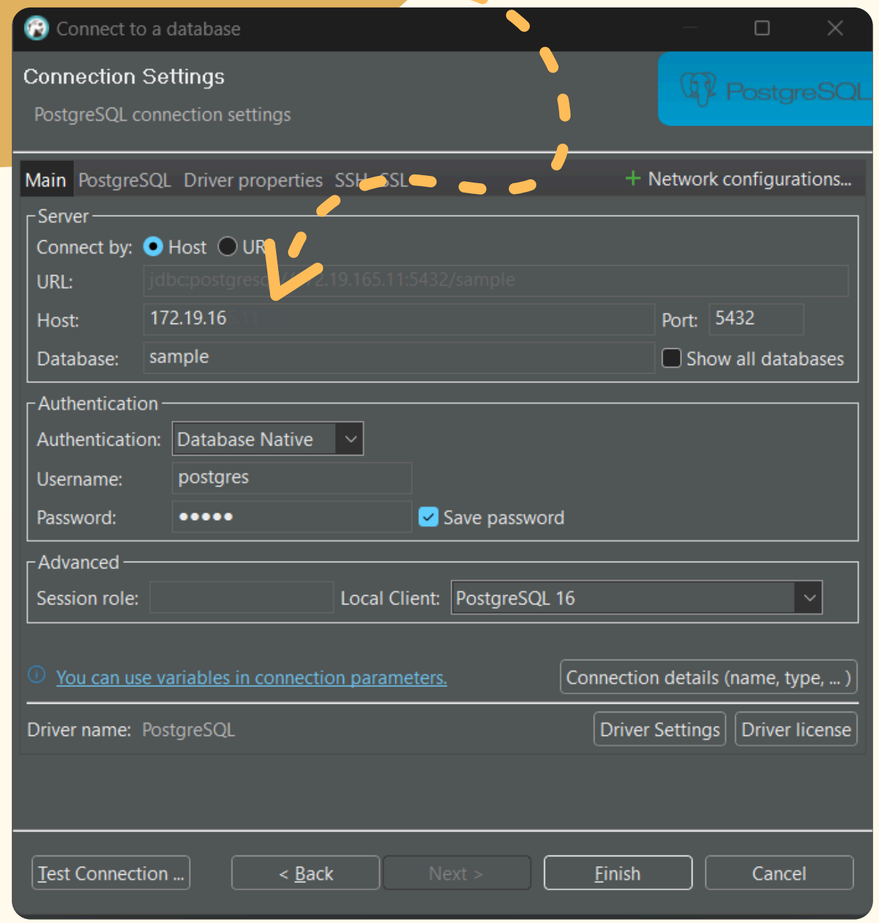
Melihat status pada docker desktop: tandanya sudah aktif

```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS                    NAMES
190e4331d792   postgres   "docker-entrypoint.s…"   28 seconds ago   Up 24 seconds   0.0.0.0:5432->5432/tcp   project-5-sib-digitalskola-postgres-db-1
```

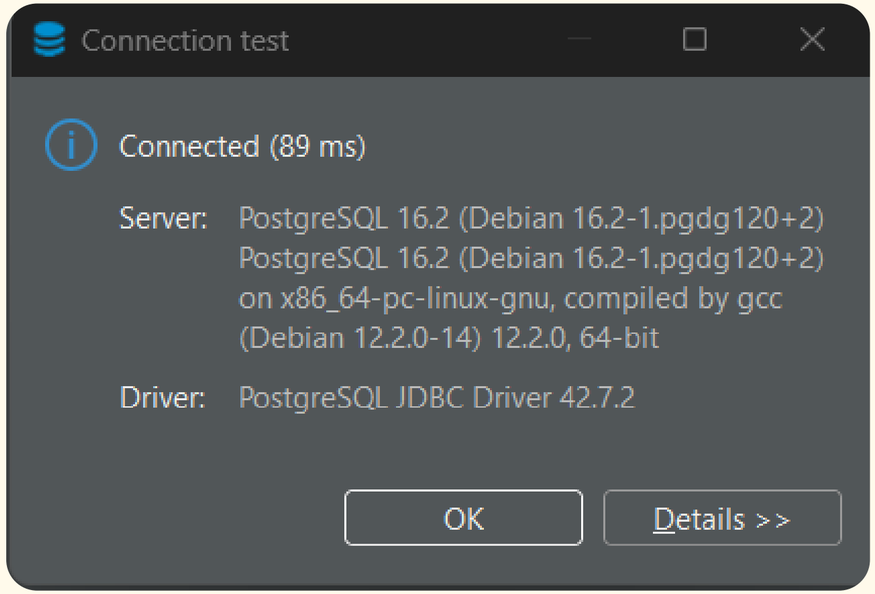
Cek status dengan command **'docker ps'**

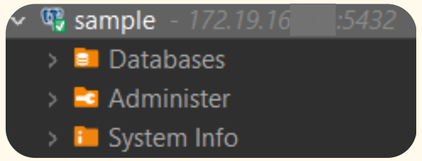
STEP 4: Test Connection Postgres

Host menggunakan IP network



Tes koneksi ke database Postgres menggunakan DBeaver





Connection tes “connected” artinya sudah berhasil terkoneksi ke Postgres dengan konfigurasi dari file docker-compose.yml

DigitalSkola
Uncover The World of Digital Skills with Us

STEP 5: Membuat Endpoint untuk Memeriksa Kesehatan Aplikasi dan Koneksi Database pada Flask

requirements.txt

```
Flask
psycopg2
python-decouple
flask-sqlalchemy
```

File requirements.txt berisikan requirements yang diperlukan.

app > main.py > ...

```python
from flask import Flask, jsonify
from decouple import config
import psycopg2


app = Flask(__name__)


@app.route("/health")
def health():
    return jsonify({"status": "oke"})


@app.route("/db_check")
def db_check():
    conn_pg = psycopg2.connect(
        host=config('MB_DB_HOST'),
        database=config('MB_DB_DBNAME'),
        user=config('MB_DB_USER'),
        password=config('MB_DB_PASS'),
        port=int(config('MB_DB_PORT')),
    )
    cur = conn_pg.cursor()
    return jsonify({"status": 200, "db": "connected"})


if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0")
```

File main.py menyediakan endpoints yang dapat digunakan untuk memeriksa kesehatan aplikasi secara umum (/health) dan untuk memeriksa koneksi ke database (/db_check)

STEP 6: Konfigurasi Dockerfile untuk Membangun Image Docker Aplikasi Flask

Dockerfile > ...

```
1   FROM python:3.7
2
3   COPY requirements.txt /requirements.txt
4
5   COPY ./app /app
6
7   WORKDIR /
8
9   RUN pip install -r requirements.txt
10
11  ENTRYPOINT ["python"]
12
13  CMD ["app/main.py", "--reload"]
```

Dockerfile bertujuan untuk membangun image Docker yang berisi aplikasi Flask beserta dependensinya, dan mengatur agar aplikasi Flask dijalankan saat container Docker berjalan.

DigitalSkola
Uncover The World of Digital Skills with Us

STEP 7: Build Image Docker

```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker build -t project-5-sib-digitalskola .
[+] Building 30.8s (9/9) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 228B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.7                                                                 1.3s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/5] FROM docker.io/library/python:3.7@sha256:eedf63967cdb57d8214db38ce21f105003ed4e4d0358f02bedc057341bcf92a0           0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 91B                                                                                              0.0s
 => CACHED [2/5] COPY requirements.txt /requirements.txt                                                                      0.0s
 => CACHED [3/5] COPY ./app /app                                                                                              0.0s
 => [4/5] RUN pip install -r requirements.txt                                                                                28.5s
 => exporting to image                                                                                                        0.7s
 => => exporting layers                                                                                                       0.6s
 => => writing image sha256:af9dc73b7c39fb6f873fe6fb983fdc1eddbb95a3f3792429fdd9b388d42b68da                                  0.0s
 => => naming to docker.io/library/project-5-sib-digitalskola                                                                 0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Command **‘docker build -t project-5-sib-digitalskola’** digunakan untuk membangun image Docker dari Dockerfile yang ada di dalam direktori.

```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
project-5-sib-digitalskola   latest    af9dc73b7c39   52 minutes ago   1.03GB
postgres                  latest    8e4fc9e18489   2 months ago     431MB
```

Mengecek docker image yang sudah dibuat dengan command **‘docker images’**

STEP 8: Menambahkan layanan flask-app pada file docker-compose.yml

docker-compose.yml

```yaml
services:
  postgres-db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: admin
      POSTGRES_USER: postgres
      PGDATA: /var/lib/postgres/data
      POSTGRES_DB: sample
    volumes:
      - pg-data:/var/lib/postgres/data
    ports:
      - "5432:5432/tcp"
  flask-app:
    image: project-5-sib-digitalskola
    environment:
      MB_DB_DBNAME: postgres
      MB_DB_HOST: postgres-db
      MB_DB_PASS: admin
      MB_DB_PORT: 5432
      MB_DB_TYPE: postgres
      MB_DB_USER: postgres
    volumes:
      - ./app:/app
    links:
      - postgres-db:postgres-db
    ports:
      - "5000:5000/tcp"
volumes:
  pg-data: #docker volume create pg-data
    external: true
```

Mematikan docker compose **'docker-compose down'** kemudian menjalankan kembali dengan command '**docker-compose up -d**'

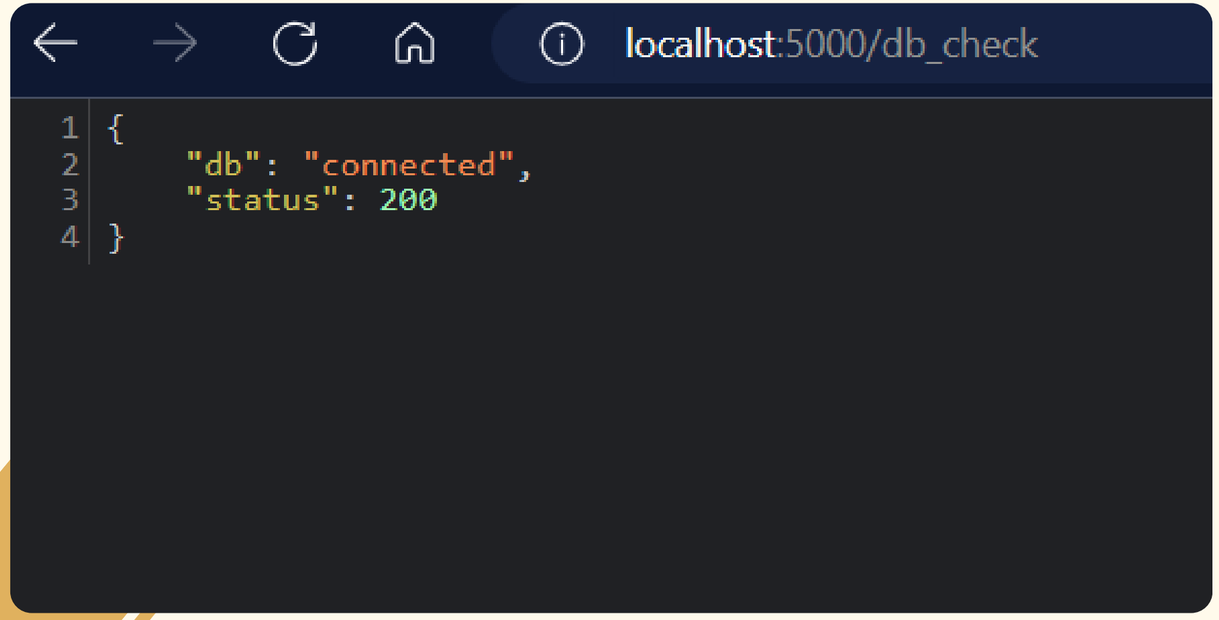
```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker-compose down
[+] Running 3/3
 ✓ Container project-5-sib-digitalskola-flask-app-1    Removed                                    0.5s
 ✓ Container project-5-sib-digitalskola-postgres-db-1  Removed                                    0.6s
 ✓ Network project-5-sib-digitalskola_default          Removed                                    0.3s
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker-compose up -d
[+] Running 3/3
 ✓ Network project-5-sib-digitalskola_default          Created                                    0.1s
 ✓ Container project-5-sib-digitalskola-postgres-db-1  Started                                    0.2s
 ✓ Container project-5-sib-digitalskola-flask-app-1    Started                                    0.1s
```

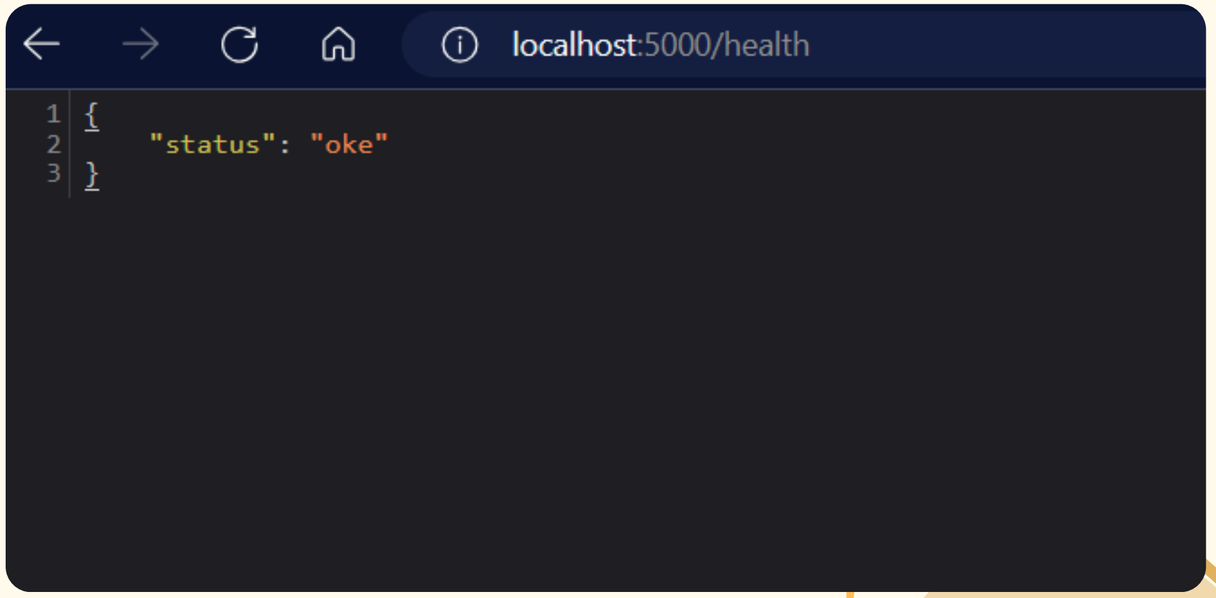
Cek status dengan command 'docker ps'

```
andrew@MSI:/mnt/d/Dokumen/TUGAS-TUGAS SIB DigitalSkola/TUGAS PROJEK (Individu + Kelompok)/PROJEK 5/PROJECT-5-SIB-DigitalSkola$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS          PORTS                    NAMES
5e59edf01ebf   project-5-sib-digitalskola     "python app/main.py …"   12 seconds ago   Up 11 seconds   0.0.0.0:5000->5000/tcp   project-5-sib-digitalskola-flask-app-1
afb8131df8b5   postgres                       "docker-entrypoint.s…"   12 seconds ago   Up 11 seconds   0.0.0.0:5432->5432/tcp   project-5-sib-digitalskola-postgres-db-1
```

DigitalSkola

STEP 9: Memeriksa Endpoint Koneksi /db_check dan /health pada localhost 5000





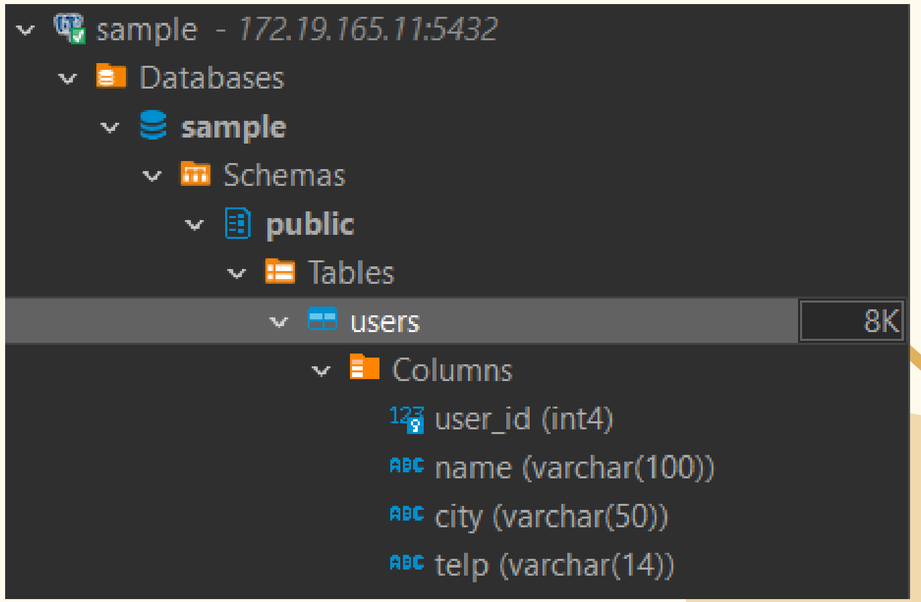
STEP 10: Membuat DDL Pada Database Postgres di DBeaver

*<sample> Script-5

```
create sequence user_id_seq
    start with 1
    increment by 1
    no minvalue
    no maxvalue
    cache 1;
CREATE TABLE public.users (
    user_id int4 NOT NULL DEFAULT nextval('user_id_seq'::regclass),
    name varchar(100) NULL,
    city varchar(50) NULL,
    telp varchar(14) NULL,
    CONSTRAINT users_pkey PRIMARY KEY (user_id)
);
```

Membuat struktur database untuk menyimpan informasi pengguna dalam proyek Flask

Terdapat tabel baru dengan nama “users” yang berisikan kolom user_id, name, city, dan telp.



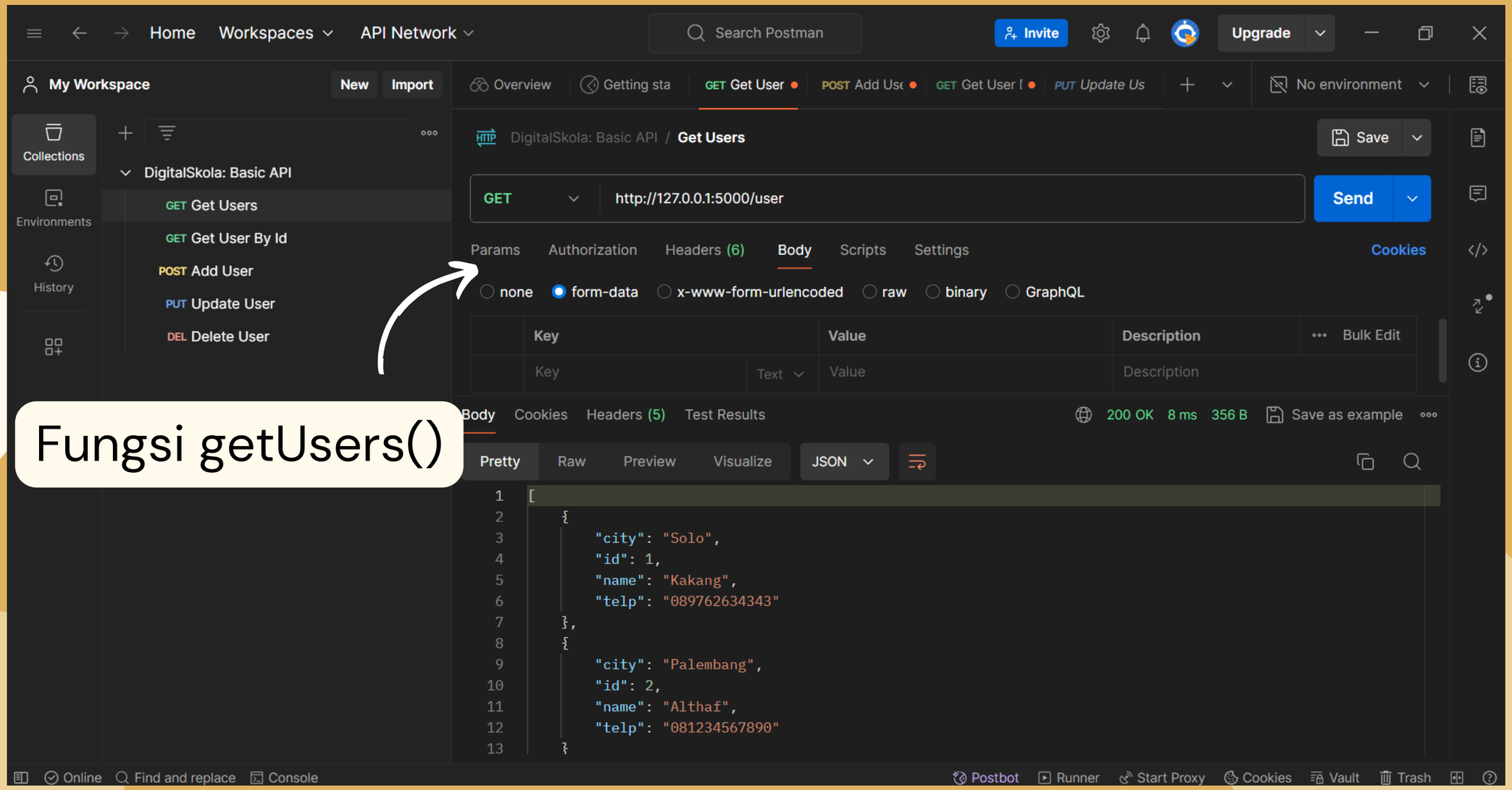
STEP 11: Penambahan Endpoint CRUD dalam Aplikasi Flask

app > main.py > ...

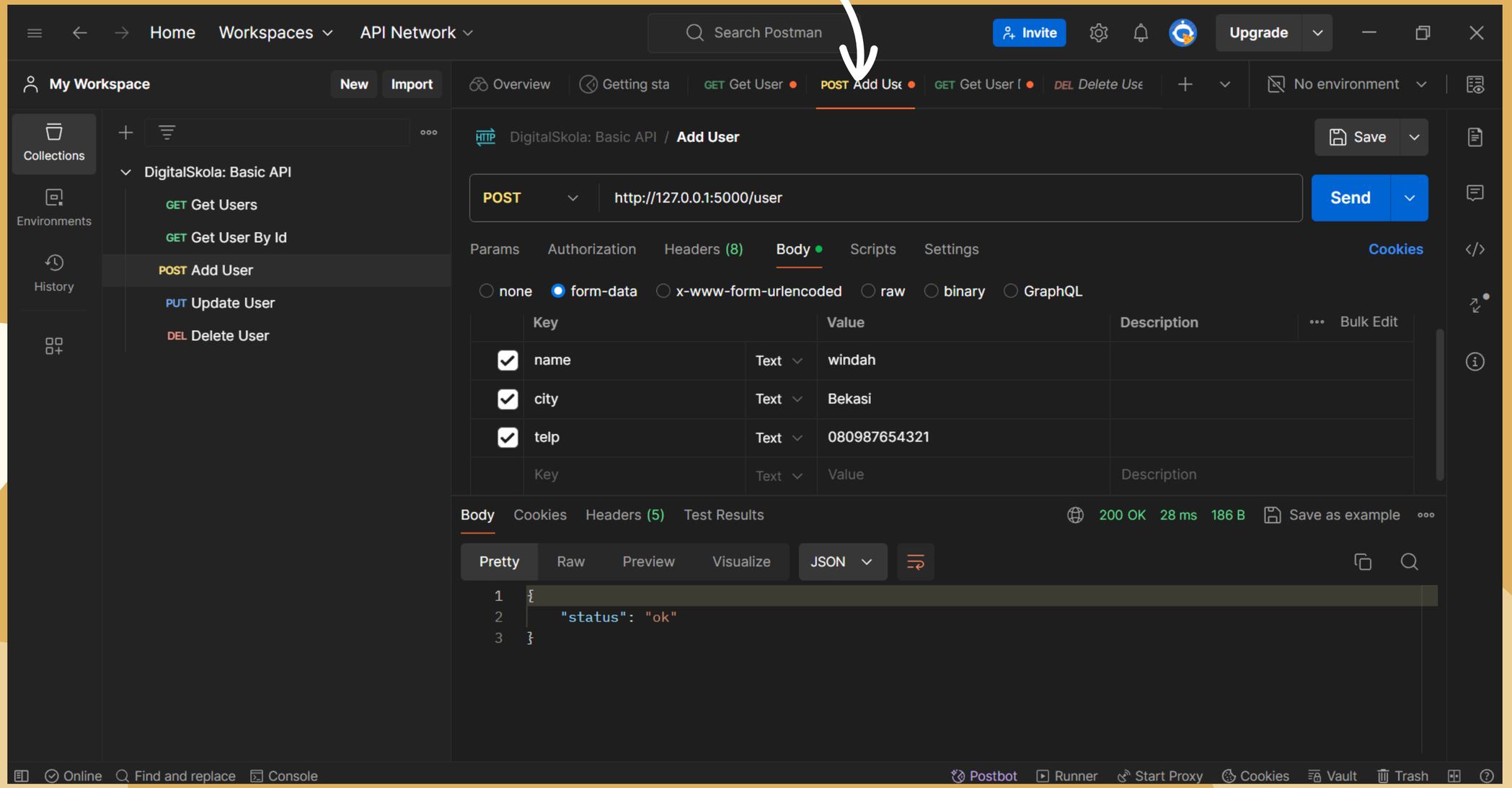
```python
from flask import Flask, jsonify, request
from decouple import config
import psycopg2
from flask_sqlalchemy import SQLAlchemy


DB_URI = f"postgresql+psycopg2://{config('MB_DB_USER')}:{config('MB_DB_PASS')}@{config('MB_DB_HOST')}:{str(config('MB_DB_PORT'))}/{config('MB_DB_DBNAME')}"
app = Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = DB_URI
app.config["SQLALCHEMY_TRACK_MODIFIACTIONS"] = True
db = SQLAlchemy(app)


class Users(db.Model):
    id = db.Column('user_id', db.Integer, primary_key=True)
    name = db.Column(db.String(100))
    city = db.Column(db.String(50))
    telp = db.Column(db.String(14))


@app.route("/health")
def health():
    return jsonify({"status": "oke"})


@app.route("/db_check")
def db_check():
    conn_pg = psycopg2.connect(
        host=config('MB_DB_HOST'),
        database=config('MB_DB_DBNAME'),
        user=config('MB_DB_USER'),
        password=config('MB_DB_PASS'),
        port=int(config('MB_DB_PORT')),
    )
    cur = conn_pg.cursor()
    return jsonify({"status": 200, "db": "connected"})


@app.route("/user", methods=["GET", "POST", "PUT", "DELETE"])
def user():
    if request.method == 'GET':
        users = Users.query.all()
        results = [{"id": u.id, "name": u.name, "city": u.city, "telp": u.telp} for u in users]
        return jsonify(results)


if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0")
```

Menambahkan fitur-fitur CRUD yang memungkinkan aplikasi untuk melakukan operasi Create, Read, Update, dan Delete pada data pengguna dalam database PostgreSQL.

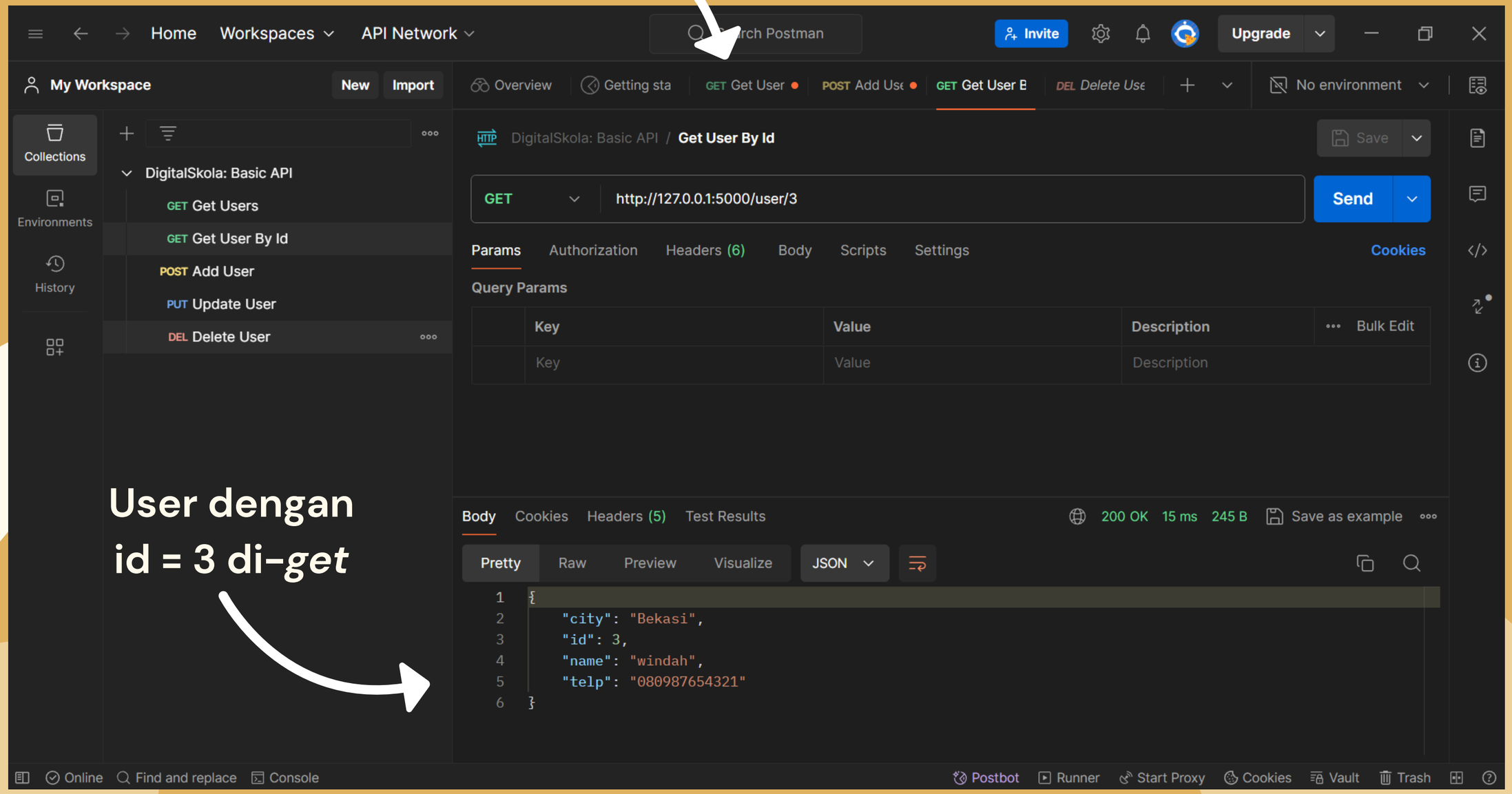
STEP 12: Menampilkan Hasil Permintaan API Flask pada Postman



Fungsi addUsers()

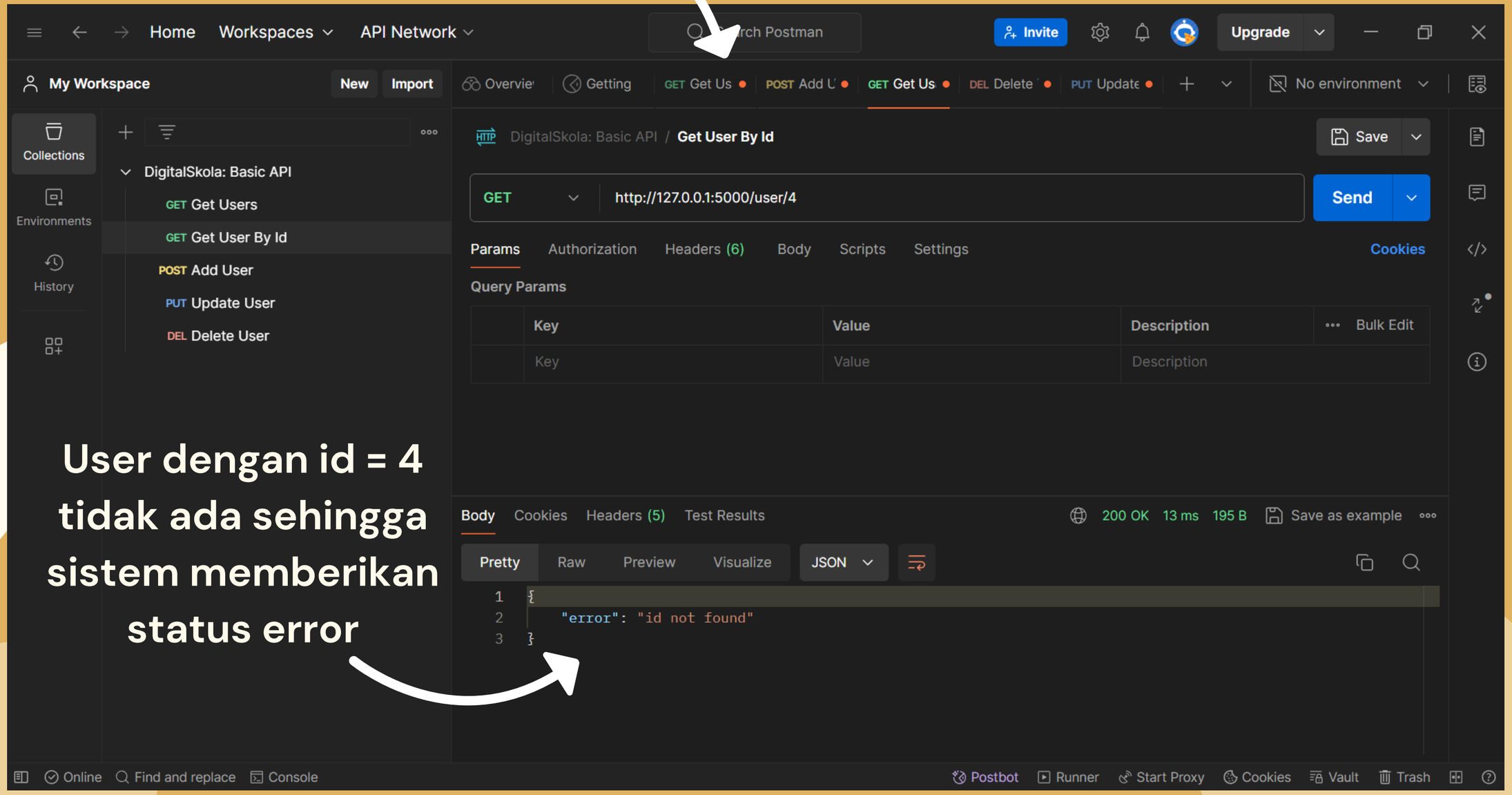


Fungsi getUserById()



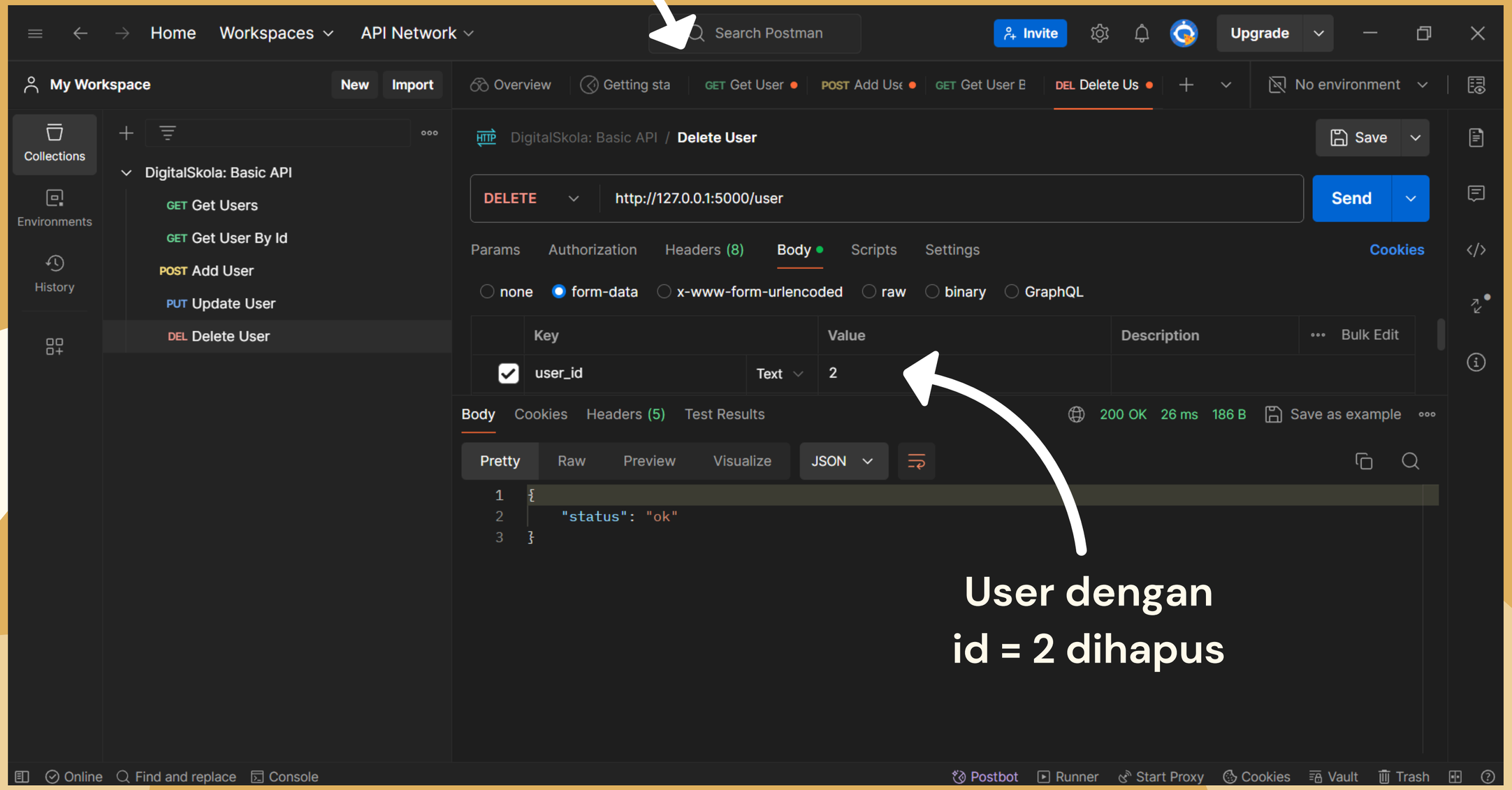
User dengan
id = 3 di-*get*

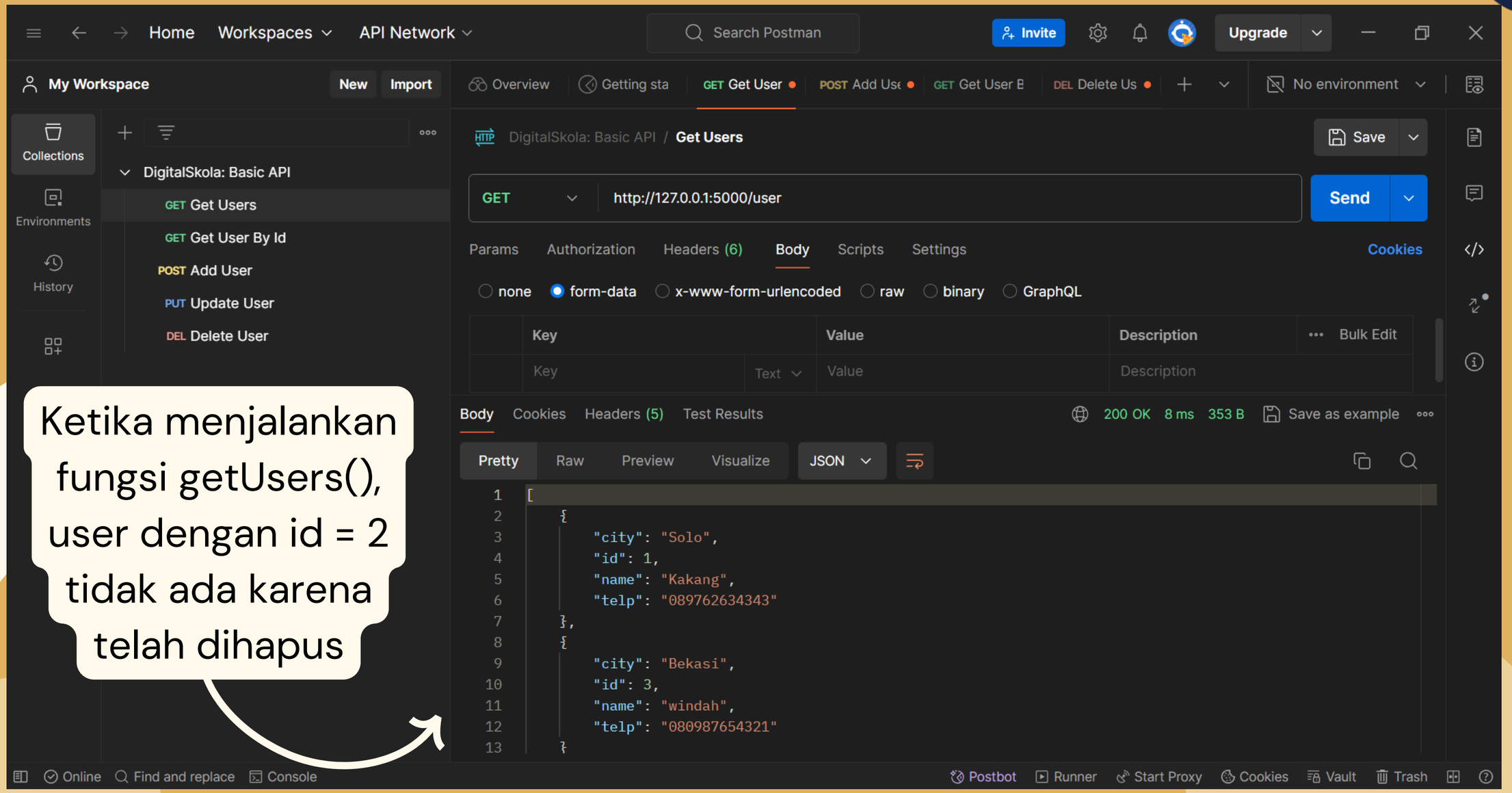
Fungsi getUserById()



User dengan id = 4 tidak ada sehingga sistem memberikan status error

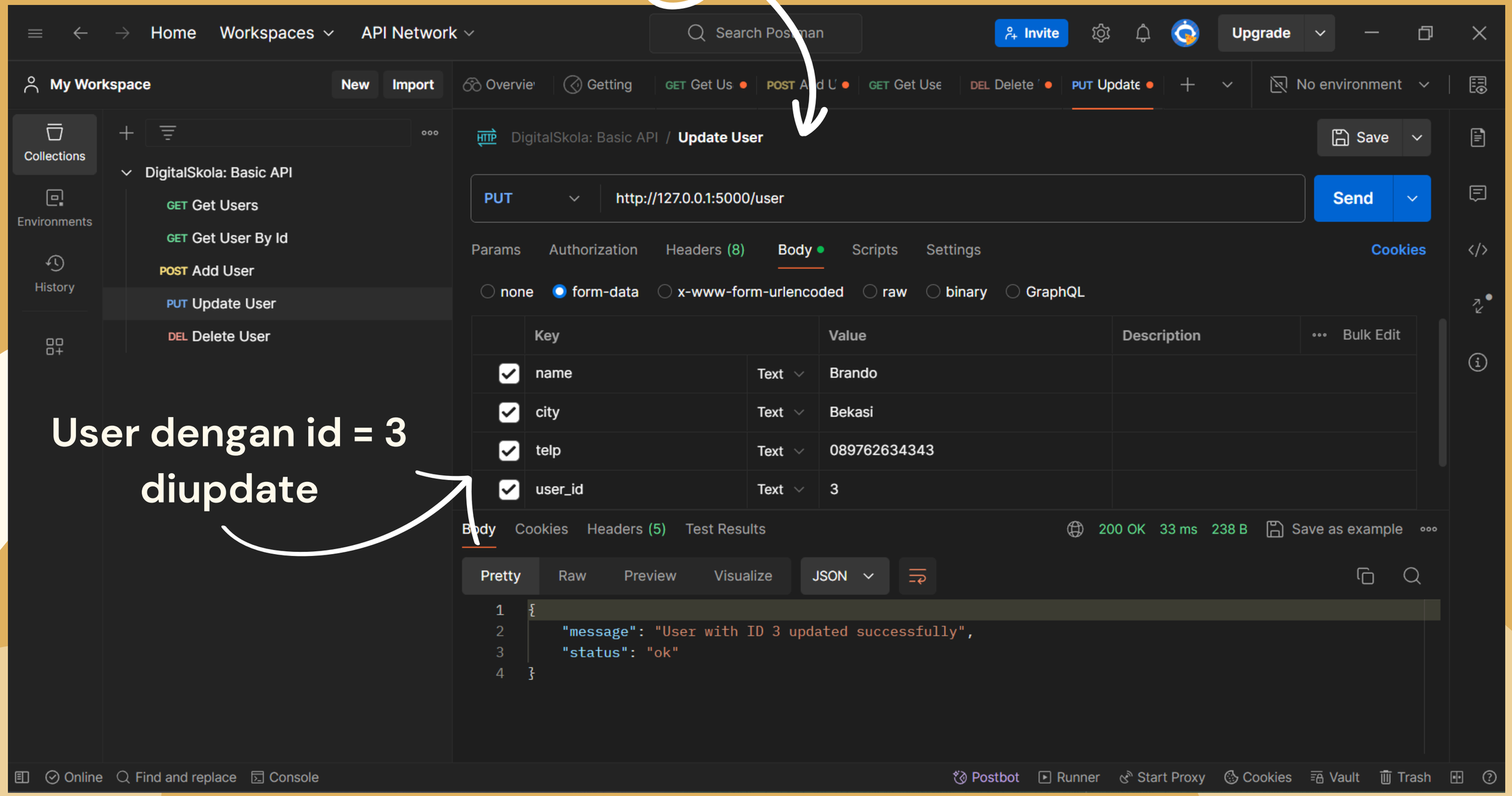
Fungsi deleteUser()



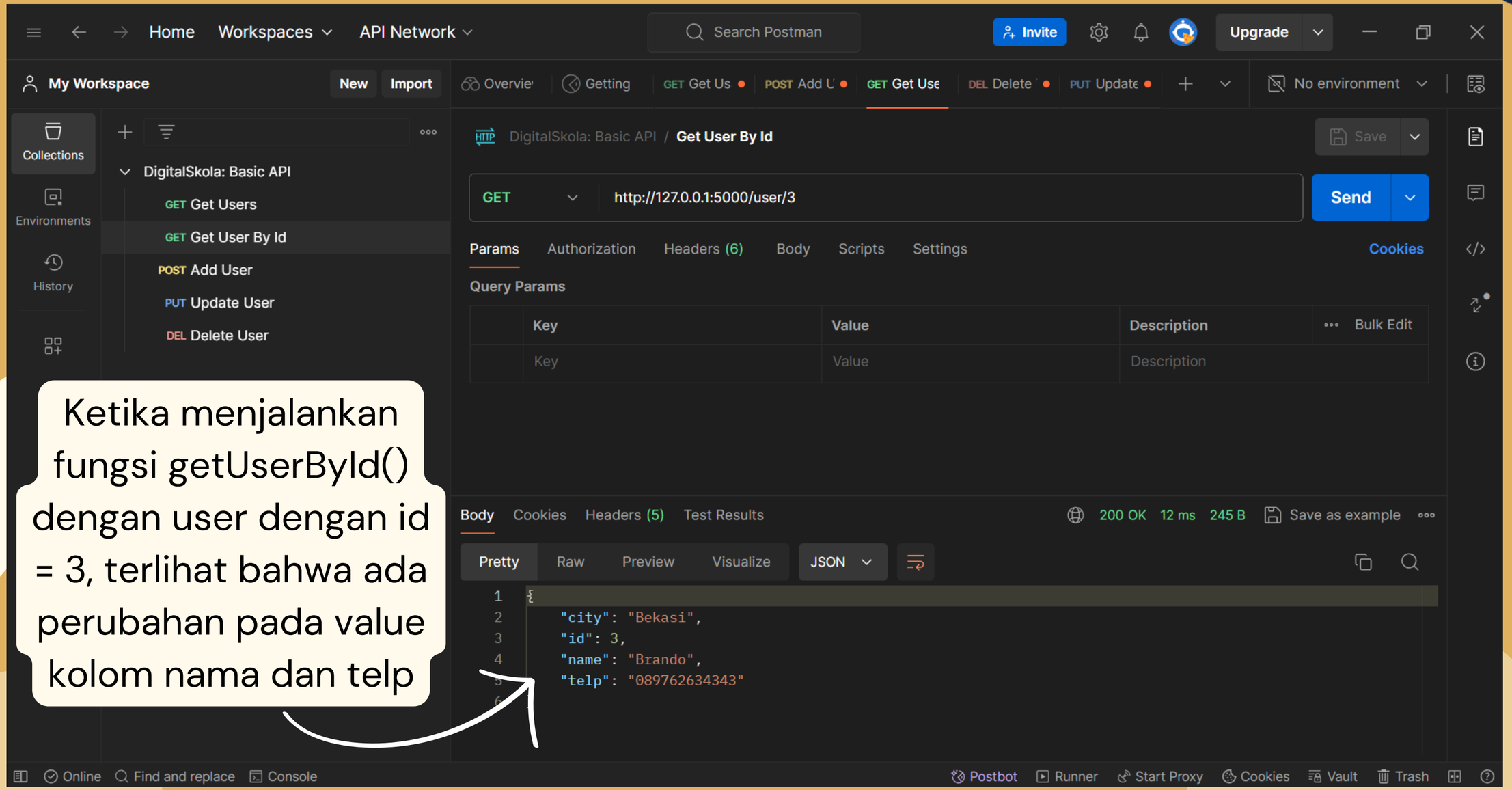


Ketika menjalankan fungsi getUsers(), user dengan id = 2 tidak ada karena telah dihapus

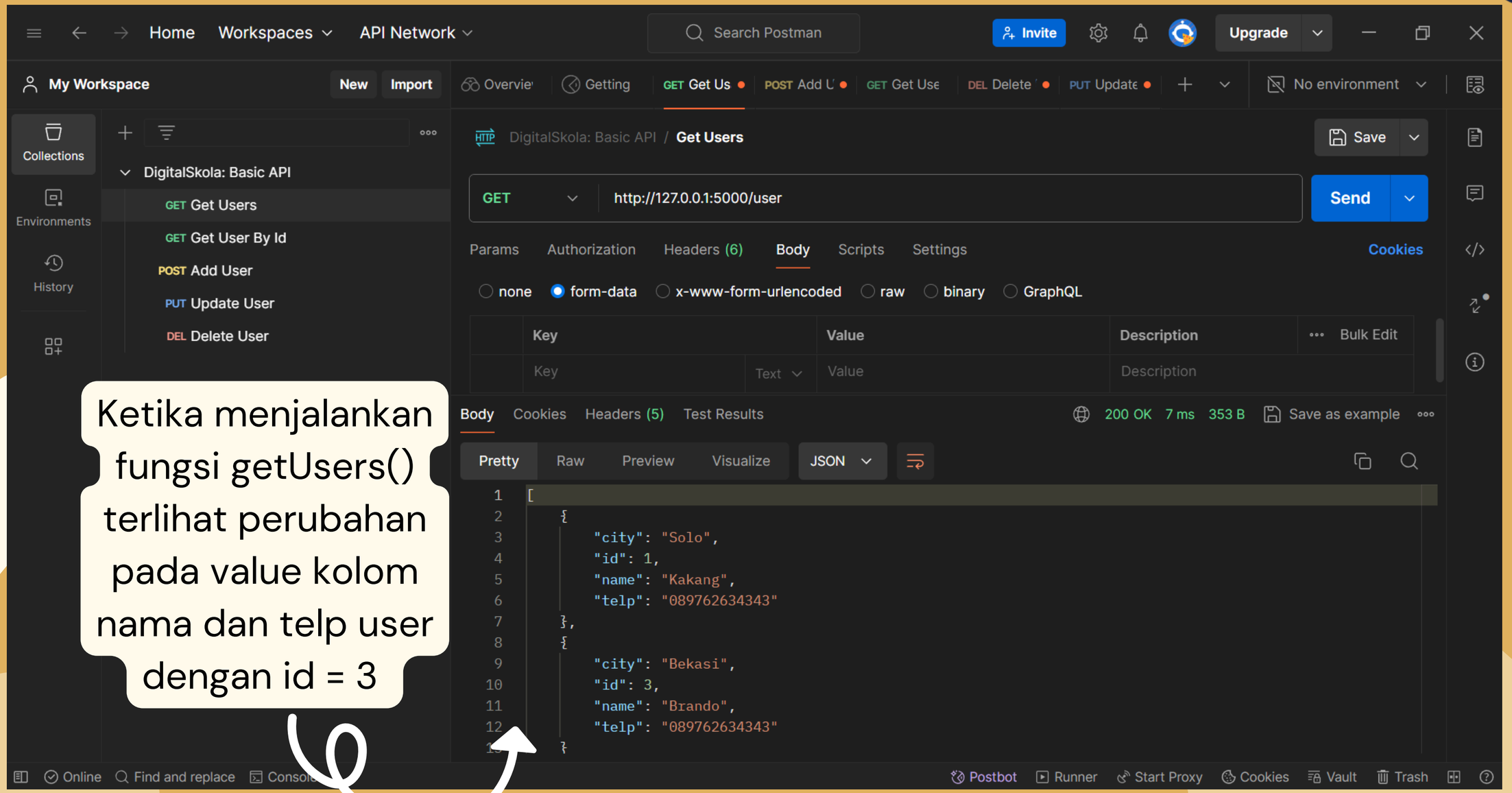
Fungsi updateUser()



User dengan id = 3 diupdate

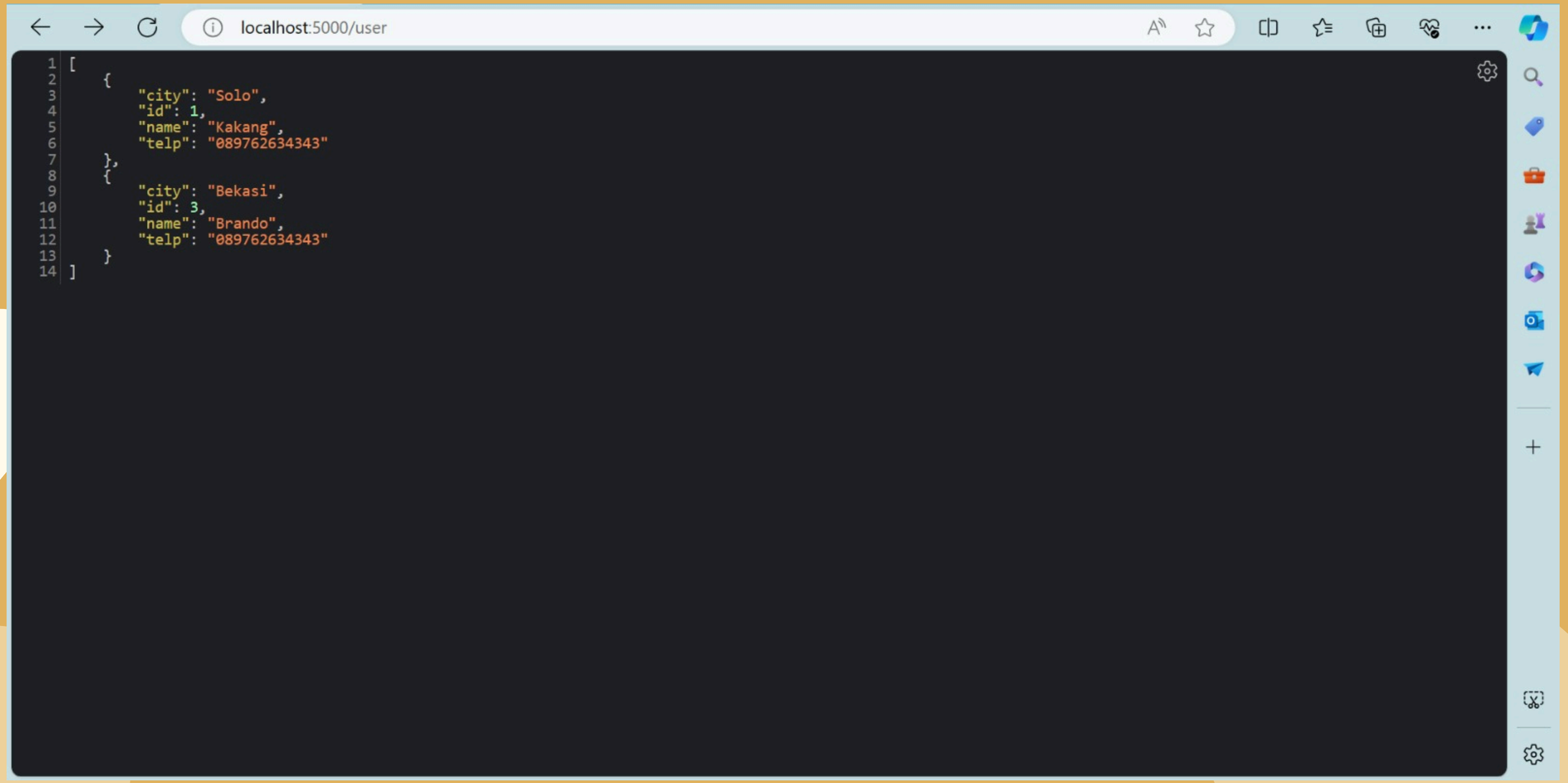


Ketika menjalankan fungsi getUserById() dengan user dengan id = 3, terlihat bahwa ada perubahan pada value kolom nama dan telp



Ketika menjalankan fungsi getUsers() terlihat perubahan pada value kolom nama dan telp user dengan id = 3

Tampilan Endpoint Koneksi /user pada localhost 5000



TERIMAKASIH