

## Tugas 2

Nama: Dimas Putra Rahardika

Nim: 220411100087

1. Jelaskan secara singkat konsep dasar REST (Representational State Transfer). Sebutkan dan jelaskan prinsip RESTful API (stateless, resource-based, client-server, cacheable, dll.). Berikan contoh skenario penggunaan REST API dalam dunia nyata.
2. Jelaskan masing-masing metode HTTP yang digunakan dalam REST API (GET, POST, PUT, DELETE, PATCH).
3. Berikan contoh endpoint REST API beserta fungsi dari masing-masing metode HTTP.
4. Status Code HTTP
  - a. Jelaskan status code HTTP yang umum digunakan dalam REST API.
  - b. Berikan contoh kasus penggunaan masing-masing status code berikut:

Jawaban

1. REST (Representational State Transfer) adalah arsitektur untuk membangun layanan web yang berskala besar. REST bukan standar, melainkan sekumpulan prinsip dan batasan yang harus diikuti untuk menciptakan API yang efisien dan fleksibel.

Prinsip-prinsip REST

### 1. Uniform Interface

Semua interaksi dengan API harus konsisten, menggunakan format standar seperti JSON dan endpoint yang jelas.

Contohnya: API pengguna dengan URL konsisten untuk operasi berbeda:

- Ambil semua pengguna: GET /users
- Tambah pengguna: POST /users
- Update pengguna: PUT /users/1
- Hapus pengguna: DELETE /users/1

### 2. Client-server

API harus memisahkan antara klien (frontend) dan server (backend) sehingga keduanya bisa dikembangkan secara independen.

Contohnya: Klien dan server bisa diperbarui atau diganti tanpa saling mempengaruhi.

### 3. Stateless

Setiap permintaan API harus bersifat mandiri, artinya server tidak menyimpan status pengguna antara permintaan.

Contohnya: Setiap permintaan harus berisi semua informasi yang diperlukan (tanpa menyimpan sesi).

### 4. Cacheable

Respons API dapat disimpan (cache) untuk mengurangi beban server dan mempercepat waktu respons.

Contohnya: respon yang terjadi berulang di simpan sehingga tidak membebani server

### 5. Layered System

API dapat memiliki beberapa lapisan (layer) untuk meningkatkan keamanan, performa, dan skalabilitas.

Contoh: API Gateway bisa memvalidasi permintaan sebelum meneruskannya ke backend.

#### 6. Code on Demand

Server dapat mengirimkan kode (misalnya JavaScript) ke klien untuk dieksekusi.

Contonya: Mengurangi beban komputasi di server dengan mengalihkan logika ke klien.

### 2. Method http REST API

1. GET (Mengambil Data). Digunakan untuk mengambil data dari server tanpa mengubahnya.
2. POST (Menambahkan Data Baru). Digunakan untuk menambahkan data baru ke server.
3. PUT (Memperbarui Seluruh Data). Digunakan untuk memperbarui seluruh data yang ada di server.
4. PATCH (Memperbarui Sebagian Data). Digunakan untuk memperbarui hanya sebagian data tanpa mengganti seluruhnya.
5. DELETE (Menghapus Data). Digunakan untuk menghapus data dari server.

### 3. Endpoint

Metode	Endpoint	fungsi
GET	/users	Mengambil daftar semua pengguna
GET	/users/{id}	Mengambil detail pengguna berdasarkan ID
POST	/users	Menambahkan pengguna baru
PUT	/users/{id}	Memperbarui seluruh data pengguna berdasarkan ID
PATCH	/users/{id}	Memperbarui sebagian data pengguna
DELETE	/users/{id}	Menghapus pengguna berdasarkan ID

### 4. Code HTTP API

#### a. Penjelasan Status Code HTTP

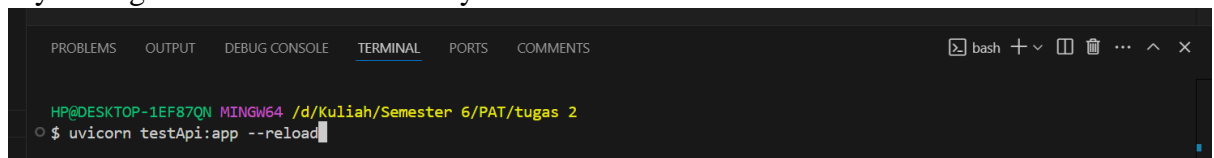
- 1xx – Informational (Informasi). Indikasi bahwa permintaan sedang diproses oleh server.
- 2xx – Success (Permintaan Berhasil). Indikasi bahwa permintaan berhasil diproses.
- 3xx – Redirection (Pengalihan). Menunjukkan bahwa klien harus mengikuti URL lain untuk mendapatkan data yang diminta.
- 4xx – Client Error (Kesalahan di Pihak Klien). Menunjukkan bahwa permintaan dari klien salah atau tidak dapat diproses.
- 5xx – Server Error (Kesalahan di Pihak Server). Menunjukkan bahwa terjadi kesalahan di server saat memproses permintaan.

b. Contoh Kasus Penggunaannya code HTTP

Status code	Arti	Contoh penggunaan
200 OK	Permintaan berhasil	Ambil daftar produk
201 Created	Data berhasil dibuat	Tambah pengguna baru
400 Bad Request	Format data salah	Kirim data dengan JSON yang salah
401 Unauthorized	Belum login	Akses profil tanpa token
403 Forbidden	Tidak memiliki izin	Akses data admin tanpa hak akses
404 Not Found	Data tidak ditemukan	Mencari produk yang tidak ada
500 Internal Server Error	Kesalahan server	Database crash
502 Bad Gateway	Server menerima respons buruk dari upstream	API Gateway gagal memproses

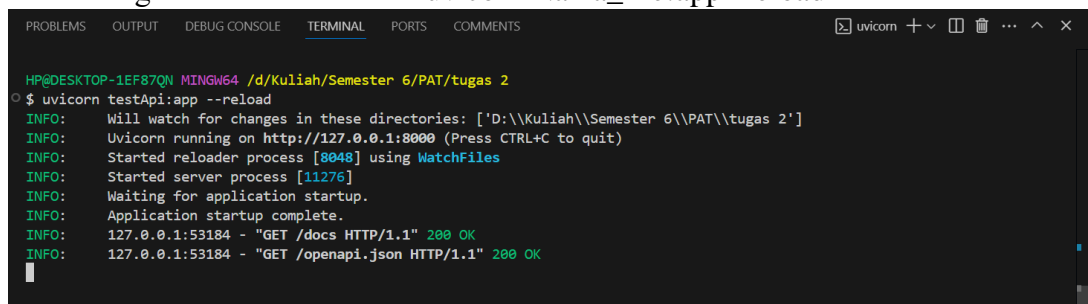
5. Dokumentasi penggunaan REST API

1. Pertama yang perlu di lakukan adalah untuk menjalankan server api. Kebetulan saya menggunakan framework API yaitu FastAPI.



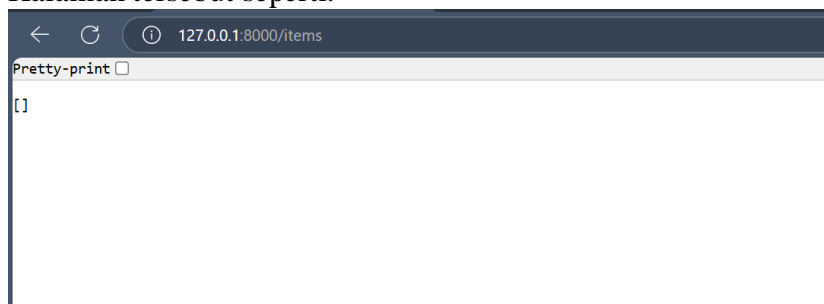
```
HP@DESKTOP-1EF87QN MINGW64 /d/Kuliah/Semester 6/PAT/tugas 2
$ uvicorn testApi:app --reload
```

Yaitu dengan cara memasukan “uvicorn Nama\_file:app –reload”



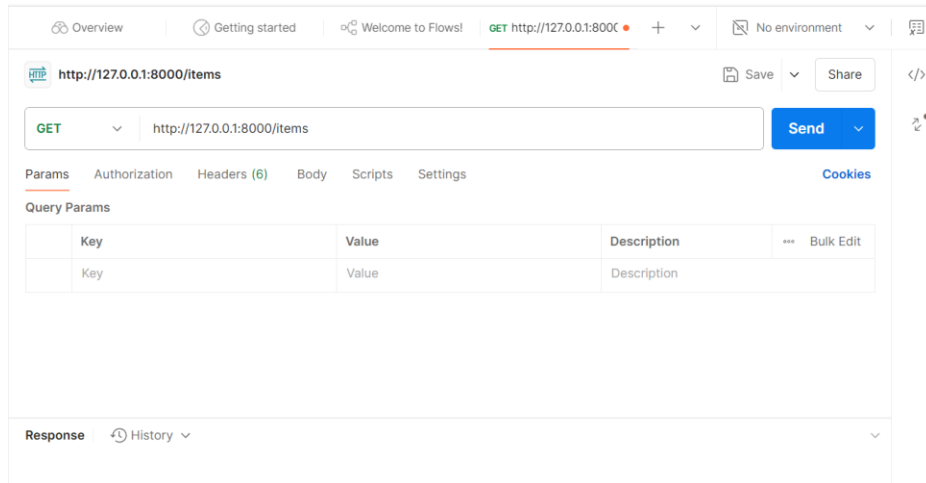
```
HP@DESKTOP-1EF87QN MINGW64 /d/Kuliah/Semester 6/PAT/tugas 2
$ uvicorn testApi:app --reload
INFO: Will watch for changes in these directories: ['D:\\Kuliah\\Semester 6\\PAT\\tugas 2']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [8048] using WatchFiles
INFO: Started server process [11276]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:53184 - "GET /docs HTTP/1.1" 200 OK
INFO: 127.0.0.1:53184 - "GET /openapi.json HTTP/1.1" 200 OK
```

Kemudian kita akan mendapatkan link berupa ip menuju halaman keluaran API. Halaman tersebut seperti:



Belum ada isinya karena data masih kosong. Halaman tersebut merupakan request data atau get data.

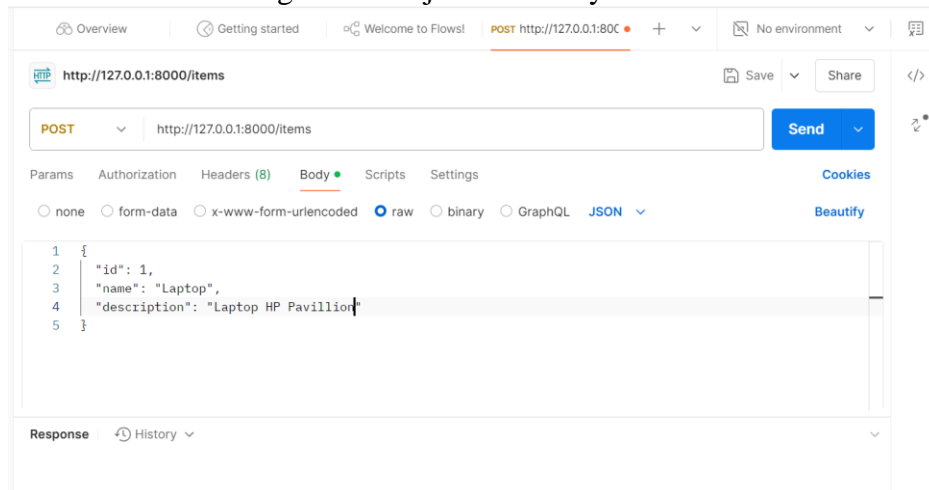
2. Untuk melakukan percobaan di sini saya menggunakan post man. Caranya adalah dengan memasukan alamat API ke postman.



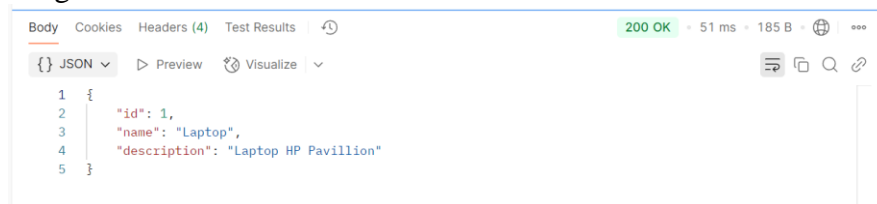
Berikut merupakan percobaan 3 endpoint get, post, delete.

a. Post (menambahkan data)

Pertama kita perlu untuk mengubah method menjadi post dan buka tab body kemudian pilih raw dan ubah bagasinya ke json, dan setelahnya kita dapat memasukan data dengan format json contohnya:



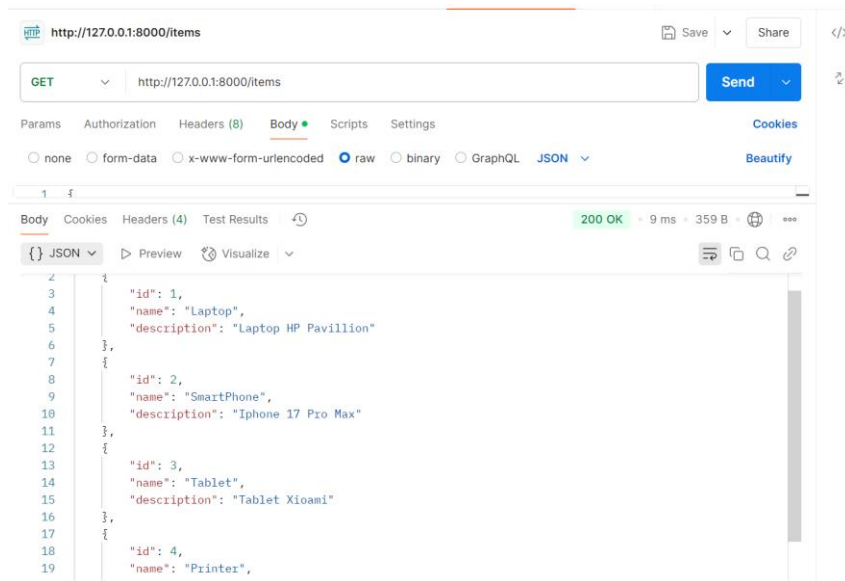
Kemudian setelahnya klik send, maka data akan di inputkan melalui api dengan status 200 ok



Saya juga akan menambahkan beberapa data lagi untuk percobaan.

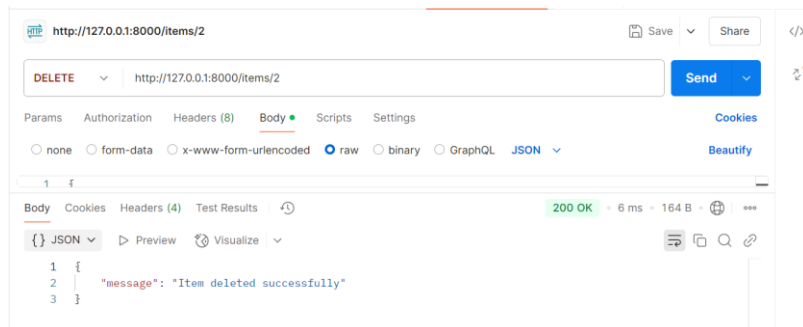
b. Get (mendapatkan data dari API)

Caranya kita hanya perlu merubah method ke get dan klik send. Dan seharusnya kita akan mendapatkan data dari api tersebut dengan status code 200 ok.

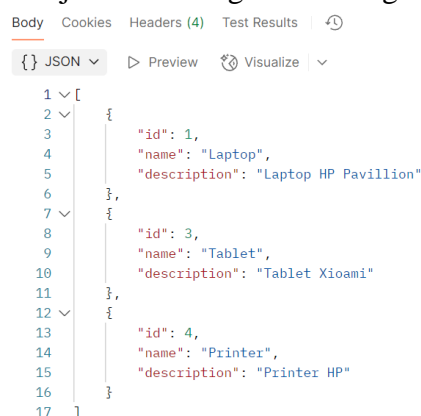


c. Delete (menghapus data di data API)

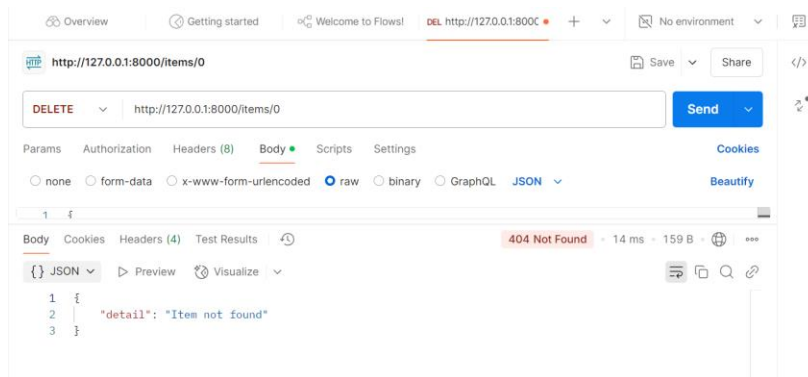
Kita perlu merubah method menjadi delete dan memasukan id dari item yang ingin kita hapus di dalam link http contohnya :” <http://127.0.0.1:8000/items/2>” kemudian kita send.



Dan jika lakukan get data dengan id no 2 akan hilang



Dan jika id item yang tidak ada saya buat pesan jika data tidak ditemukan dengan code 404 data not found



3. link repository: <https://github.com/dimasmahardika/implementsi-REST-API.git>