

1. URL merupakan kepanjangan dari Uniform Resource Locator yang berarti suatu rangkaian karakter yang memiliki format dengan standar tertentu serta dapat digunakan untuk petunjuk alamat sebuah sumber yang ada di Internet.
- REST API memiliki standarisasi dalam pemakaian URL. Dengan menggunakan URL yang sama kita dapat melakukan 5 operational berdasarkan HTTP Verbs.

D. Get : untuk membaca sebuah record atau daftar record

ex : `http://localhost/books`

D. Post : untuk ~~menghapus~~ menambah sebuah record

ex : `http://localhost/books`

Content : { title, authors, ... }

D. Put : untuk mengupdate semua field dalam sebuah record

ex : `http://localhost/books/1`

(mengupdate buku yang memiliki ID 1)

D. Delete : untuk menghapus sebuah record

ex : `http://localhost/books/2`

(menghapus buku yang memiliki ID 2)

2. Penggunaan request method berdasarkan HTTP Verbs yaitu GET, POST, PUT dan DELETE. Untuk kode respon ketika menggunakan REST :

D. 200 OK → Respon ini menunjukkan bahwa permintaan berhasil dibuat (sukses)

D. 400 bad request → permintaan jelek. Hal ini terjadi jika POST dan PUT tidak lulus validasi atau format salah.

D. 3xx → Redirect

D. 500 Internal Server Error → Menunjukkan kegagalan karena keadaan yang tak terduga.

D. 1xx → Status Informasi

3.) Axios adalah sebuah library open source yang saat ini booming untuk melakukan request HTTP karena memiliki banyak kelebihan.

D. GET

```
const URL = 'http://jsonplaceholder.typicode.com/posts';
```

```
axios.get(URL)
```

- pemanggilan hanya mengirimkan alamat URL endpoint

D. POST

Perbedaan saat melakukan post, kita menambahkan properti data untuk data yang ingin dikirim ke server. Axios menggunakan promise.

4.) D. Grant type Authorization Code

↳ Pada grant type ini, kita akan menggunakan generate code yang berasal dari authorization server untuk ditukarkan dengan sebuah token, dimana token ini kita gunakan untuk mengakses sebuah resource atau API.

D. Grant type password

↳ Pada grant type ini, kita akan menggunakan username dan password langsung dari ownernya. Grant type ini biasanya digunakan jika aplikasi yang dibangun merupakan aplikasi pribadi atau pembuat aplikasi berasal dari perusahaan yang sama. Contohnya adalah ketika kita login pada aplikasi Facebook.

5.) Arsitektur monolit merupakan sebuah arsitektur dalam pembuatan aplikasi dimana semua komponen menjadi satu kesatuan. Komunikasi akan komunikasi kepada aplikasi atau service lain bisa jadi tidak ada, karena aplikasi ini telah mencakup seluruh kode yang dibutuhkan. Sedangkan microservices adalah arsitektur yang memecahkan kepada banyak service atau aplikasi yang tidak terikat erat dengan yang lainnya. Microservice juga lebih fleksibel dan cepat, deployment dapat dilakukan terpisah untuk masing-masing services. Lalu keuntungan menggunakan microservice yaitu :

- Mudah untuk kode suatu service. Karena suatu service hanya terfokus pada satu fungsi, kode yang di butuhkan.
- Deployment lebih cepat dan ringan. Karena kita dapat melakukan deployment hanya pada satu service tertentu tanpa terikat service lain.
- Ke salahan bisa di isolasi. Karena ketika 1 service bermasalah, service lain tidak terganggu.
- Tidak terikat pada satu teknologi.