

# API JSON

Buatlah struktur JSON dari website E-Commerce, JSON berisi menu, submenu dan daftar barang yang ada pada web.

Misal :

Struktur website youtube.com dalam JSON

```
1 {~
2   "main": [~
3     {~
4       "title": "Home",~
5       "url": "https://www.youtube.com",~
6       "content": [~
7         {~
8           "title": "Javascript Best Practices",~
9           "url": "https://youtube.com/watch?v=abcdefghijkl",~
10          "likes": 1500,~
11          "dislikes": 5,~
12          "description": "lorem ipsum dolor sit amet, consectetur adipiscing elit",~
13          "comments": [~
14            {~
15              "name": "Alice",~
16              "comment": "Very good",~
17              "likes": 53,~
18              "dislikes": 9,~
19              "replies": 5~
20            }~
21          ]~
22        }~
23      ]~
24    },~
25    {~
26      "title": "Trending",~
27      "url": "https://www.youtube.com/feed/trending",~
28      "content": [~
29        {~
30          "title": "Javascript Best Practices",~
31          "url": "https://youtube.com/watch?v=abcdefghijkl",~
32          "likes": 1500,~
33          "dislikes": 5,~
34          "description": "lorem ipsum dolor sit amet, consectetur adipiscing elit",~
35          "position": 1,~
36          "comments": [~
37            {~
38              "name": "Alice",~
39              "comment": "Very good",~
40              "likes": 53,~
41              "dislikes": 9,~
42              "replies": 5~
43            }~
44          ]~
45        },~
46        {~
47          "title": "ReactJS Tutorial",~
48          "url": "https://youtube.com/watch?v=abcdefghijkl",~
49          "likes": 1534,~
50          "dislikes": 1,~
51          "description": "lorem ipsum dolor sit amet, consectetur adipiscing elit",~
52          "position": 2,~
53          "comments": [~
54            {~
55              "name": "Alice",~
56              "comment": "Very good",~
57              "likes": 59,~
58              "dislikes": 1,~
59              "replies": 20~
60            }~
61          ]~
62        }~
63      ]~
64    }~
65  ]~
66 }~
67
```

# REACT JS

Buatlah Project dengan react, yang berisi halaman list postingan dan profil yang di baca dari JSON. Desain halaman sesuai selera anda.

Hint:

1. Buat Projek dengan React.js, blog bisa diubah dengan nama projek yang anda suka.

```
npx create-react-app blog
```

2. Masuk ke projek dan pasang react-router-dom (yarn add/npm install) pada proyek anda.
3. Buat folder bernama pages di dalam folder src, serta isikan berkas profile.js , baca.js, blog.js .
  - a. Blog.js

```
import React from 'react'

class Blog extends React.Component{
  render(){
    return(
      <h1>Blog</h1>
    )
  }
}

export default Blog;
```

- b. Profile.js

```
import React from 'react'

class Propil extends React.Component {
  render(){
    return <h1>Propile</h1>
  }
}

export default Propil;
```

c. Baca.js

```
import React from 'react'

class Baca extends React.Component{
  render(){
    const { params } = this.props.match;
    return(
      <>
        <h1>Judul</h1>
        <p>{params.id}</p>
      </>
    )
  }
}

export default Baca
```

4. Buat folder bernama data di dalam folder src, serta isikan berkas data.js.
- a. Buat Array berisi obyek : Blog dan Profile, sesuaikan value dari elemen yang ada dengan keinginan anda.

```
let Blogs = [
  {
    "id": 0,
    "judul": "Judul",
    "deskripsi": "Deskripsi disini",
    "konten": "Isi nya disini apa ya UwU"
  },
  {
    "id": 1,
    "judul": "Judul1",
    "deskripsi": "Deskripsi Disini",
    "konten": "Isinya disini apa"
  }
];

let Profile = {
  "name": "Nama Anda",
  "about": "Kamu siapa?",
  "github": "https://github.com/"
};
```

- b. Kemudian buat fungsi mempermudah akses data (controller-like)

```
export function allBlog(){
    return Blogs
}

export function fetchBlog(id){
    return Blogs.map((data)=>{
        if(data.id == id){
            return data
        }
    })
}

export function profiles(){
    return Profile
}
```

5. Kemudian import Blog, Profile dan react-router-dom.

```
import App from './App';
import Blog from './pages/blog';
import Propil from './pages/profile';
import Baca from './pages/baca';

import * as serviceWorker from './serviceWorker';
import { Route, Link, BrowserRouter as Router } from 'react-router-dom';
```

- a.

- b. Lalu hubungkan komponen dengan alamat URL (Routing)

```
ReactDOM.render(  
  <Router>  
    <div>  
      <ul>  
        <li>  
          <Link to="/">Home</Link>  
        </li>  
        <li>  
          <Link to="/profile">Profile</Link>  
        </li>  
        <li>  
          <Link to="/blog">Blog</Link>  
        </li>  
      </ul>  
      <Route path="/" component={App} />  
      <Route exact path="/profile" component={Propil} />  
      <Route exact path="/blog" component={Blog} />  
      <Route path="/blog/:id" component={Baca} />  
    </div>  
  </Router>  
,  
  document.getElementById('root')  
) ;
```

Gunakan fungsi pada data.js untuk query data pada JSON , kemudian tampilkan data menjadi tampilan komponen UI , UI = f(data).

Contoh hasil yang diinginkan :

- **Profile**



**Yoghaswara Hadi Nugroho**

GIT INTEGRATION  
OgataKaede

Contact

Back To Home Sites

**About me :**

Unavailable yet!


**Nothing :**

TBD

TBD

TBD

- **Blog**



**Yoghaswara Hadi Nugroho**

Beranda




**Golang dan GraphQL**

Pratik membuat GraphQL Server menggunakan Golang

Nothing
TBD
TBD
TBD

## • Baca



**Yoghaswara Hadi Nugroho**

Beranda

## Golang dan GraphQL

**GraphQL** (*Graph Query Language*) merupakan konsep API dengan cara memasukan kueri untuk digunakan mengambil dan mengubah data pada sisi server. GraphQL dengan sifat pengguna di kueri tidak terhubung dengan basis data secara langsung, ini umumnya pemahaman keliru saat mengenal GraphQL. GraphQL bisa berhubungan dengan basis data baik relasional, berbasis dokumen, graf dan lainnya.

Seperti pada sifat API pada umumnya yaitu menghubungkan permintaan(*request*) dengan fungsi pembalas(*response*), GraphQL mendefinisikan kueri yang nantinya akan di isi oleh fungsi pembalas. Bila dilihat dari sudut pandang pemrogram berorientasi objek, GraphQL merupakan interface yang kita akan implementasi pada model/fungsi pembalas (bebas dengan bahasa pemrograman bagian backend).

Berikut contoh kueri dan kembalianya:

- Kueri

Nothing
TBD
TBD
TBD