Membuat Dokumentasi API dengan Format Yaml

“Janganlah pernah menyerah ketika kamu masih mampu berusaha lagi. Tidak ada kata berakhir sampai kamu berhenti mencoba” - Brian Dyson

Main Objective

- Mengenal apa itu swagger
- Mengenal tools/jenis swagger
- Mengenal fungsi swagger
- Mengenal apa itu Api
- Mengenal jenis-jenis Api
- Mengenal arsitektur Api
- Mengenal komponen swagger
- Mengenal komponen pada object paths
- Mengetahui step by step pembuatan dokumentasi Api
- Mengetahui cara test Api pada swagger

Apa itu Swagger?

Swagger adalah *tools* yang digunakan untuk mendokumentasikan API atau web service yang dibangun.

Ada 3 jenis dokumentasi API, yaitu adalah sebagai berikut :

- **API Documentation**

Dokumentasi API yang diperuntukkan kepada pengguna agar mudah dibaca dan dipahami.

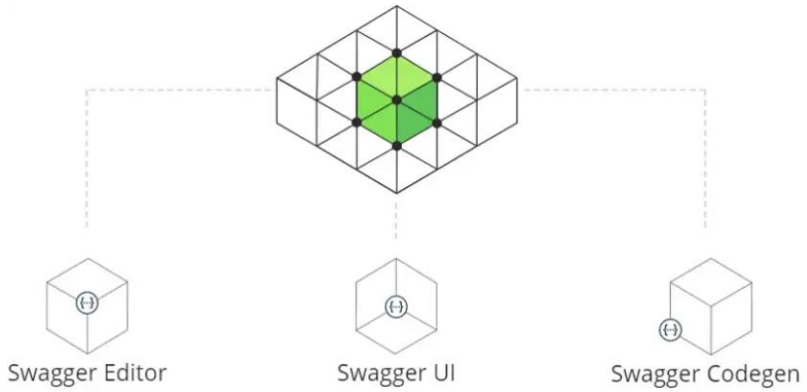
- **API Specification**

Lebih terstruktur serta programmable yang menjelaskan bagaimana suatu API tertentu bekerja, karakteristik serta output yang akan dihasilkan.

- **API Definition**

Dokumentasi API yang ditujukan untuk mesin agar dapat diotomatisasi serta dieksekusi di server.

Tools Swagger



Sumber : https://swagger.io/tools/

Swagger Editor
Editor API untuk mendesain API dengan OpenAPI Specification dan dapat disimpan dalam format yaml atau json.

Swagger UI
UI interaktif yang dapat digunakan di browser untuk memvisualisasikan definisi OpenAPI Specification.

Swagger Codegen
Code generator yang dapat digunakan untuk membuat server stubs dan client SDKs secara langsung dari OpenAPI Specification yang telah kita buat.

Fungsi Swagger

1. API Design
   Swagger membuat desain API menjadi lebih mudah dengan menggunakan *tools* yang mudah digunakan oleh pengembang.
2. API Development
   Swagger menyediakan *tools* untuk membuat prototipe dan membangun fungsionalitas API dengan cepat.
3. API Documentation
   Swagger dapat membuat dokumentasi serta memvisualisasikan dan memelihara dokumentasi API.
4. API Testing
   Swagger dapat digunakan untuk pengujian fungsional, keamanan dan kinerja dari OpenAPI Specification. *Tools* Swagger memudahkan pembuatan, pengelolaan dan pelaksanaan pengujian API dengan cepat.
5. API Mocking and Virtualization
   Swagger dapat mengurangi ketergantungan dalam siklus pengembangan dan pengujian, serta tiruan realistis dan dinamis dengan menggunakan Swagger dan ReadyAPI Virtualization.
6. API Governance
   Swagger dapat mengelola program API secara bersama dengan tim yang terdiri dari lima orang atau lebih.
7. API Monitoring
   Swagger dapat memantau ketersediaan, kecepatan dan fungsionalitas API mulai dari Spesifikasi OpenAPI (OAS) dengan AlertSite untuk OAS. Swagger dapat memonitoring dan lacak perilaku API secara instan untuk memastikan semuanya berjalan dengan lancar.
8. OpenAPI & Swagger
   Swagger menyediakan *tools* yang mudah digunakan untuk memanfaatkan kemampuan Spesifikasi OpenAPI (OAS).

Apa itu API?

API merupakan singkatan dari *Application Programming Interface* yaitu merupakan sebuah *interface* yang menghubungkan antara satu aplikasi dengan aplikasi lainnya. API memperlihatkan sekumpulan data dan fungsi untuk memfasilitasi antar aplikasi serta memungkinkan untuk bertukar informasi. API digunakan sebagai penerjemah komunikasi antara *client* dengan *server*. API bermanfaat untuk memudahkan membangun aplikasi yang fungsionalitas dan kompleks serta dapat mengembangkan aplikasi menjadi lebih efisien.

Jenis-jenis Api

Berdasarkan pengunaannya, terdapat empat jenis API sesuai dengan hak aksesnya yaitu adalah sebagai berikut:

1. Public API

Public API sering disebut juga dengan Open API yang akan menjadi fokus pembahasan kita kali ini. API jenis ini dapat digunakan oleh siapa saja dalam lintas platform yang berbeda.

2. Private API

Private API tidak dapat digunakan secara umum, biasanya API jenis ini dibuat untuk penggunaan internal dalam pengembangan aplikasi tertentu.

3. Partner API

Partner API dapat digunakan secara umum tetapi hanya sebatas pihak yang sudah memiliki izin penggunaannya.

4. Composite API

Composite API merupakan API yang dapat menyimpan data-data dari berbagai server.

Arsitektur Api

API dibangun menggunakan arsitektur tertentu, terdapat beberapa Arsitektur API yaitu adalah sebagai berikut:

- RPC (Remote Procedure Call)

RPC merupakan arsitektur API yang digunakan untuk membantu kinerja dari client side dan server side dalam hal komunikasi dengan menggunakan konsep yang sederhana. RPC memiliki dua jenis berdasarkan media perpindahan datanya yaitu XML-RPC dan JSON-RPC.

- SOAP (Simple Object Access Protocol)

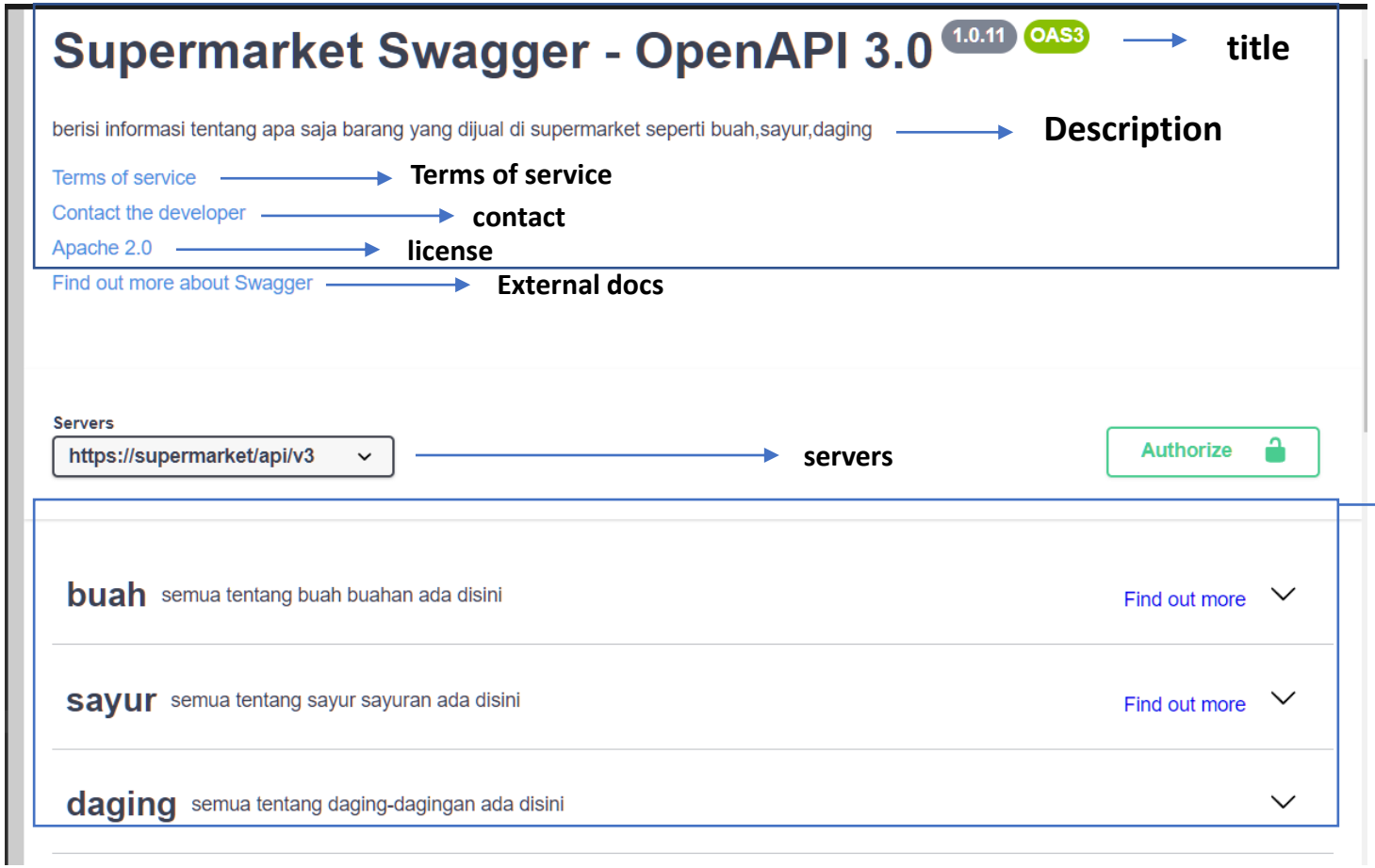
SOAP meruapakan arsitektur API yang dimana data yang digunakan berupa XML.

- REST (Representational State Transfer)

REST merupakan arsitektur API yang sangat popular pada saat ini. Kelebihan dari REST yaitu memiliki bentuk data berupa JSON sehingga lebih ringan serta mudah dalam proses pengembangannya.

Komponen Swagger

Swagger memiliki object untuk mendokumentasikan API, berikut adalah beberapa object swagger



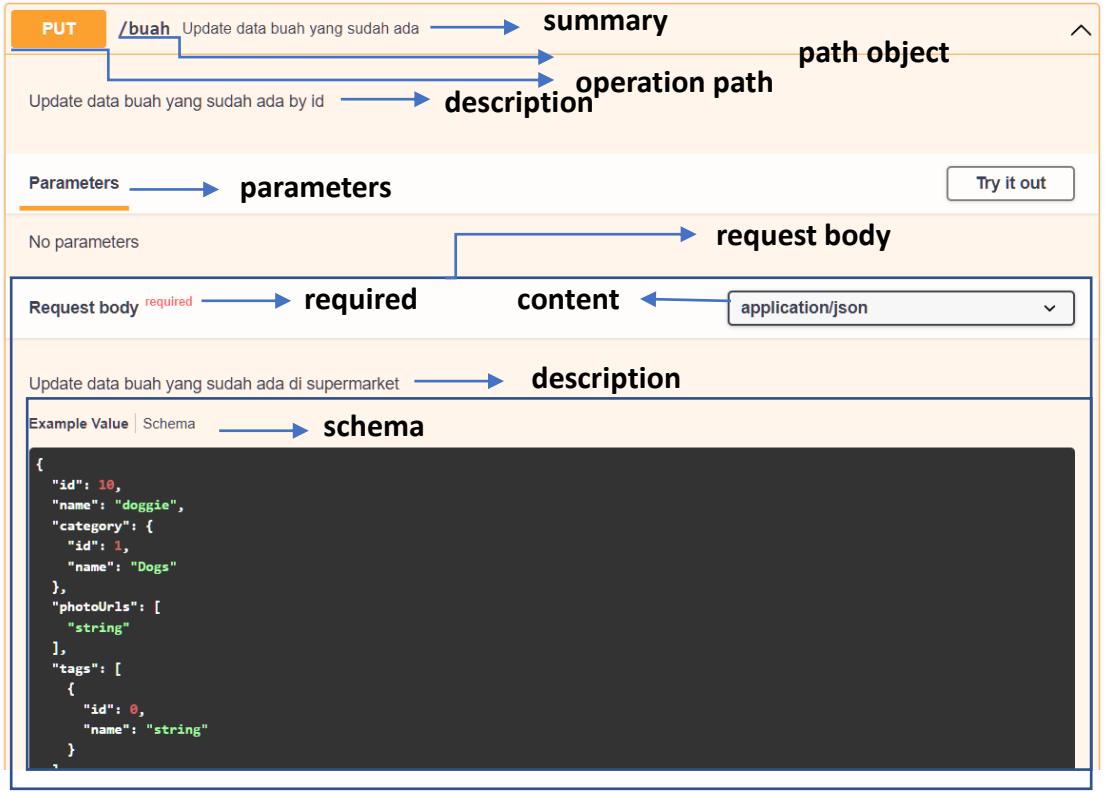
Keterangan dari setiap komponen/object openapi swagger:

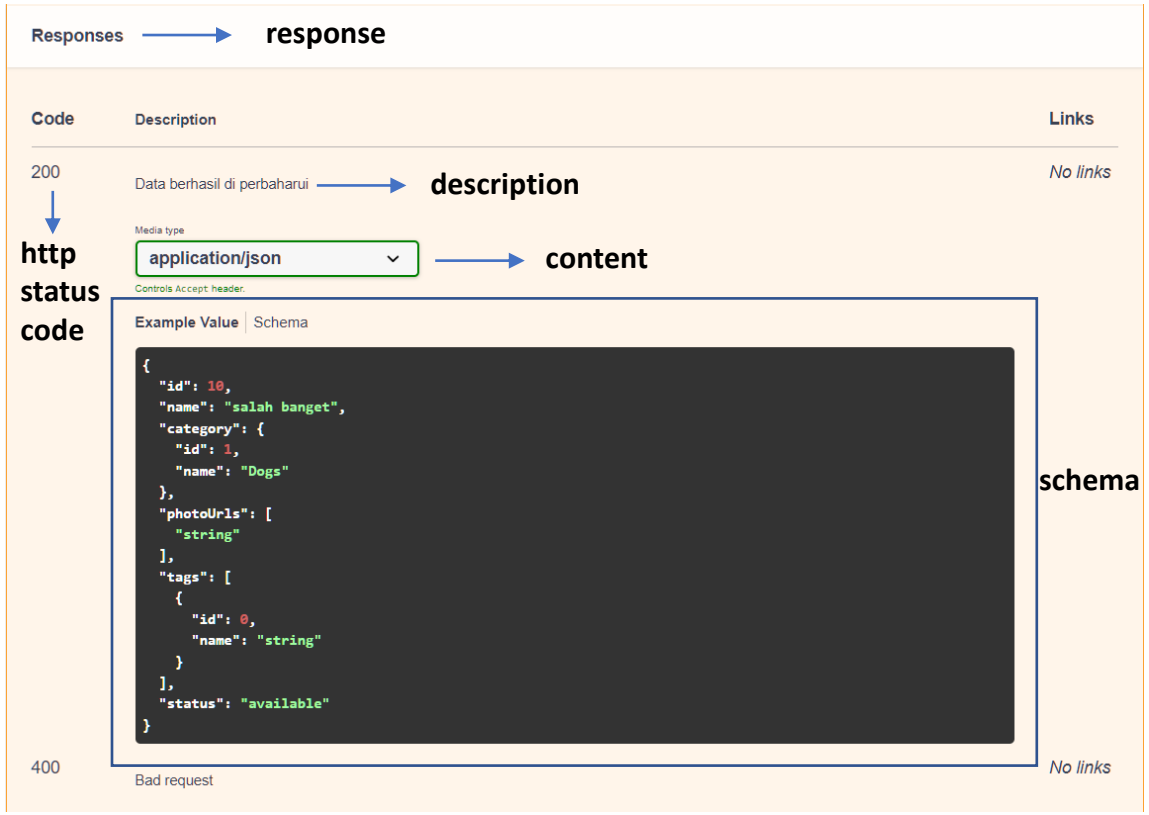
1. **info**: Penjelasan tentang dokumentasi swagger yang dibuat yang isinya ada **title** swagger, **description** dari swagger tersebut,**termsOfService** s&k pakai api tersebut, **contact** yg bisa dihubungi, **license** api tersebut serta **version** api tersebut
2. **externalDocs:** info tambahan diluar dari object openapi info
3. **servers:** informasi url yang digunakan oleh api
4. **tags:** grouping/pengelompokan api, contoh pada gambar disamping ada 3 grup api yaitu buah, sayur, daging
5. **paths:** untuk membuat url directory api

contoh: /buah(artinya url/path akan masuk ke dalam tags buah)

Komponen pada paths

Didalam object path memiliki object/komponennya tersendiri diantaranya:





Keterangan komponen paths

operation path: semacam http method karena operasi ini terdiri dari get,put,post,patch,delete dan lain-lain

summary: ringkasan dari url path

path object: sebuah url yang digunakan untuk pengelompokkan operation path. Contoh: kita membuat path objek dengan nama **/buah** dan **/buah/{buahId}**, setiap kita membuat method yang urlnya mengandung **/buah** maka method tersebut masuk ke dalam grup url **/buah.** Dalam satu grup url **/buah** bisa mempunyai satu atau lebih dari satu method seperti get,post,put.

description: deskripsi dari path yang dibuat

parameters: bersifat optional tergantung kebutuhan, biasanya digunakan pada method GET dan memiliki komponen/objeknya sendiri. Misal id, nama

request body: request data yang diperlukan API, bersifat optional tergantung kebutuhan, biasanya digunakan pada method POST,PUT,PATCH. Memiliki objeknya sendiri seperti:

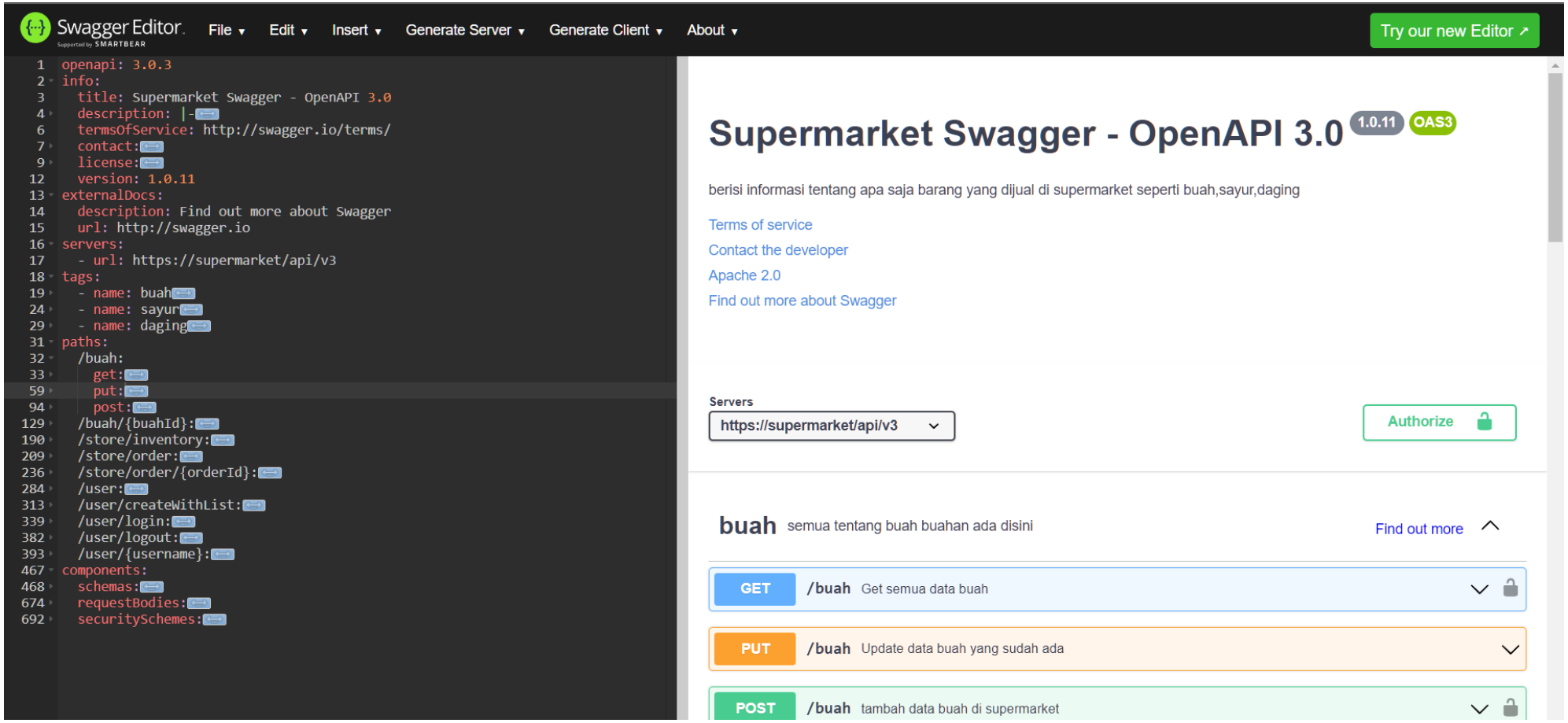
- **Content** : bentuk/konten data pada request body tapi yang paling familiar adalah application/json dan application/xml
- **Required**: untuk memvalidasi request body wajib diisi atau tidak boleh kosong. Required bersifat optional digunakan tergantung kebutuhan
- **Description**: deskripsi dari request body
- **Schema**: skema atau contoh request data yang diperlukan API

response: hasil balikan yang dikirim API dari request yang sudah dilakukan. Response memiliki komponennya sendiri diantaranya:

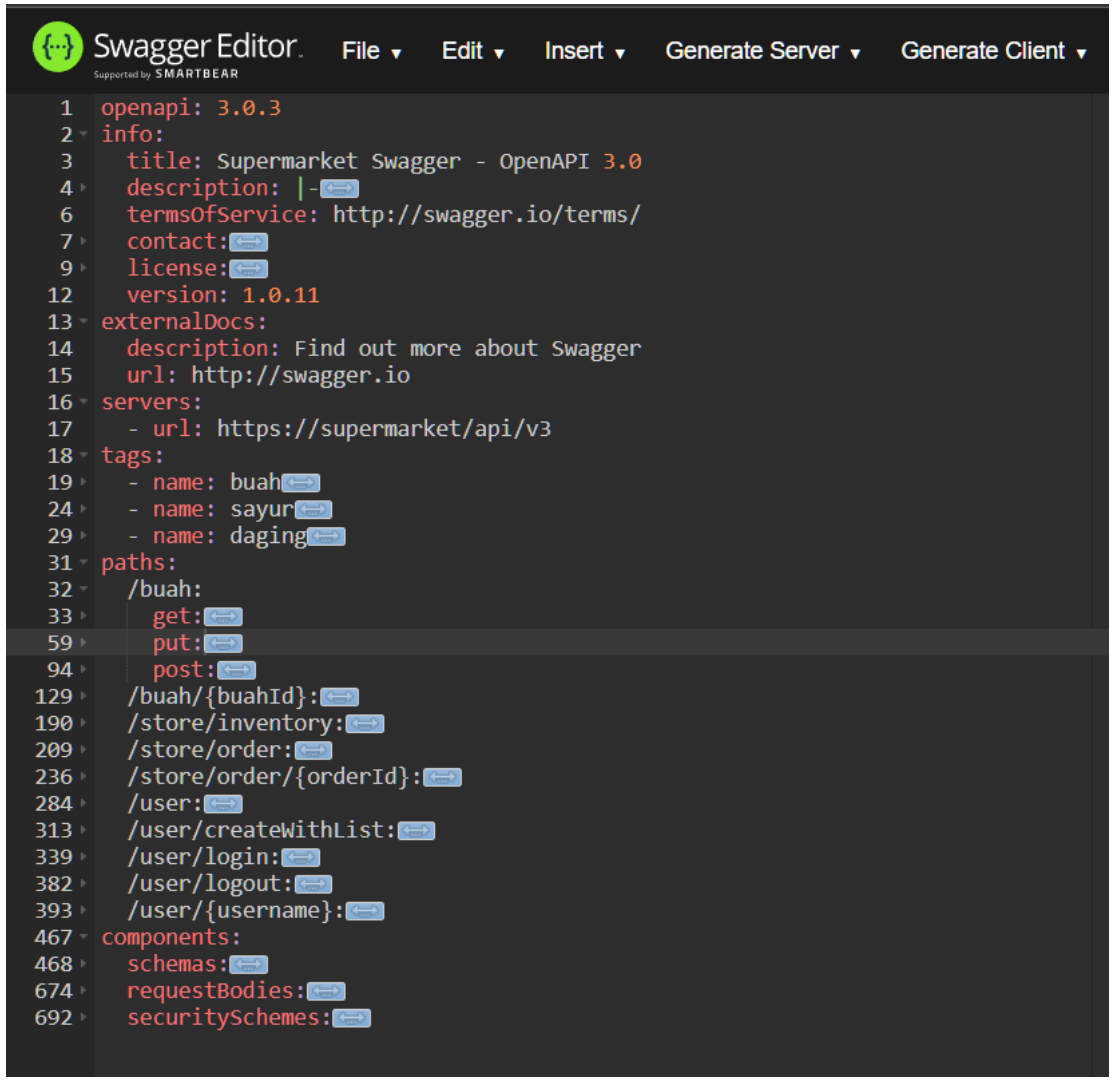
- http status code: kode yang digunakan sebagai informasi dari status request yang dilakukan kepada API. Contohnya 200 jika statusnya berhasil
- Description: deskripsi dari response
- Content: bentuk/konten data pada response tapi yang paling familiar adalah application/json dan application/xml
- Schema: skema atau contoh hasil response data dari API

Cara membuat dokumentasi Api di Swagger

1. Buka https://editor.swagger.io/. tampilan awalnya akan seperti ini. Sudah ada contoh yang disiapkan oleh swagger dalam bentuk yaml



2. Kemudian kita bisa langsung membuat dokumentasi API. Disini kita akan membuat dokumentasi API supermarket swagger, memiliki 3 kategori yaitu buah,sayur dan daging. Contohnya seperti gambar dibawah ini



openapi: wajib diisi, isinya adalah versi openapi latest

info: wajib diisi, sebagai informasi mengenai api yang kita dokumentasikan seperti judul,deskripsi,contact person,dll

externalDocs: bersifat optional, digunakan jika kita ingin membuat informasi tambahan

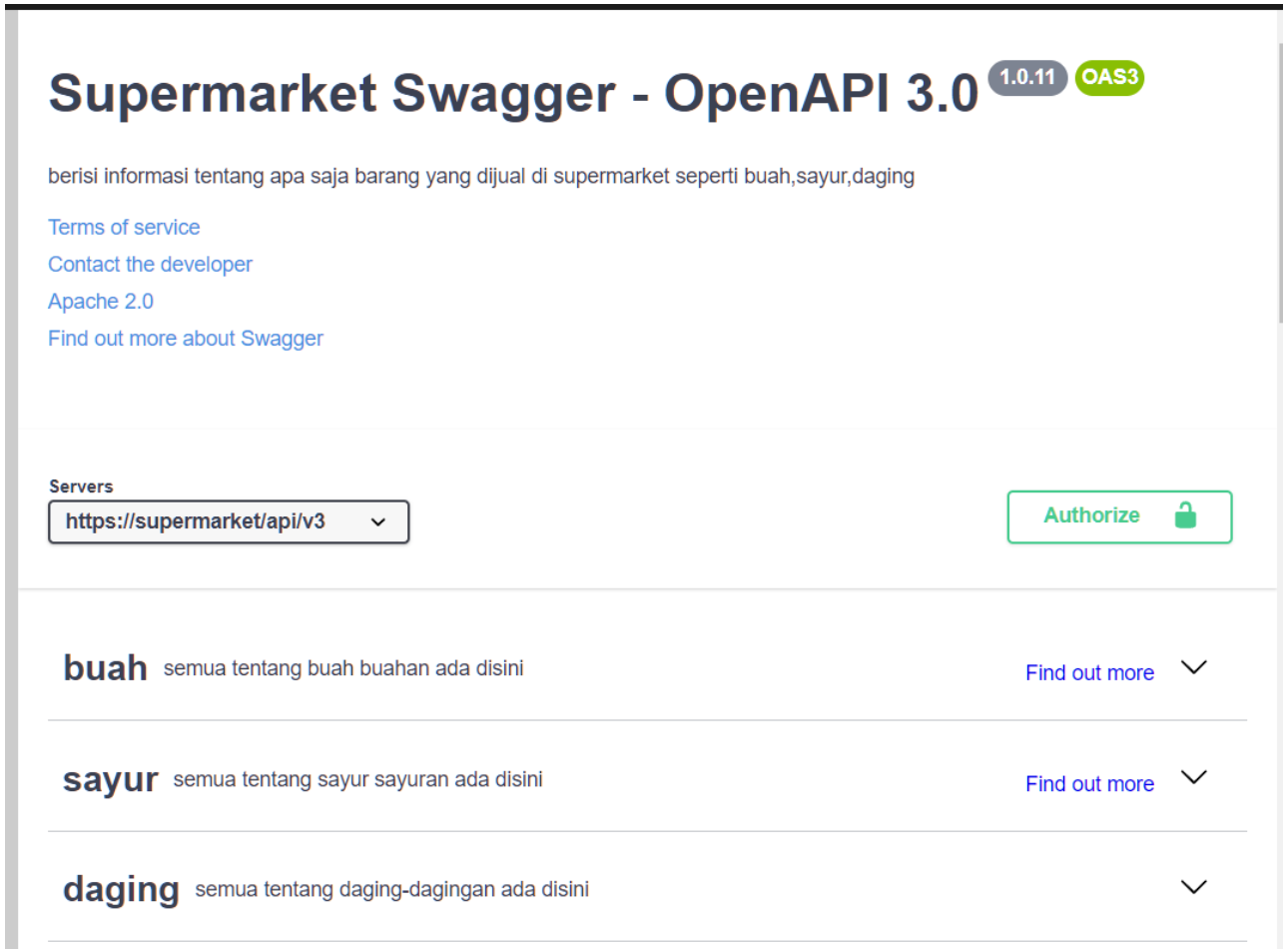
servers: digunakan sebagai tempat URL Api yang akan digunakan

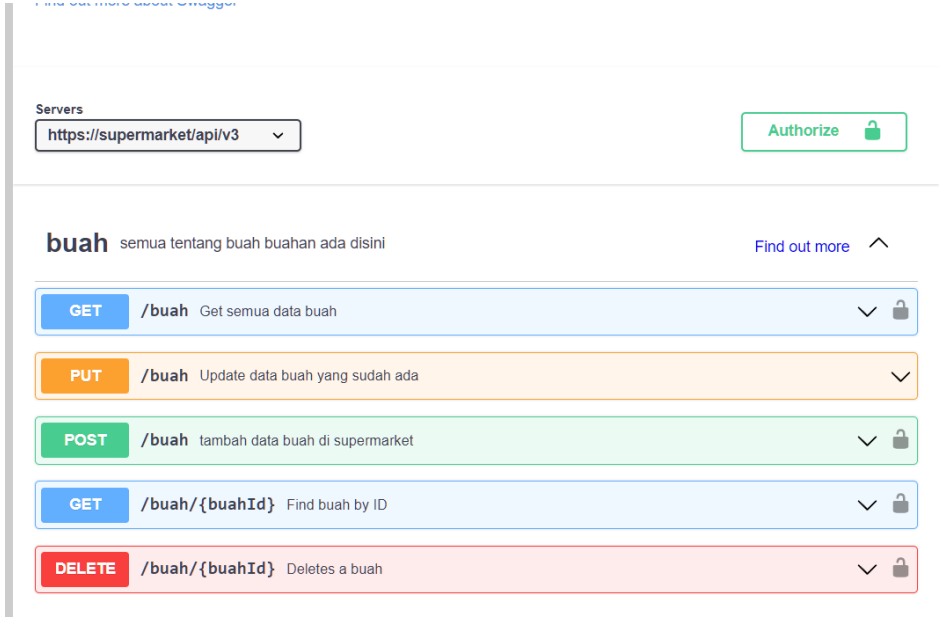
tags: grouping/pengelompokan api, contoh pada gambar disamping ada 3 grup api yaitu buah, sayur, daging

paths: untuk membuat url directory api contoh: /buah(artinya url/path akan masuk ke dalam tags buah)

components: digunakan sebagai tempat untuk membuat skema, request body, security schema yang bisa digunakan oleh objek lain dan objek komponen itu sendiri

3. Jika sudah, kita bisa run dan hasilnya akan seperti ini

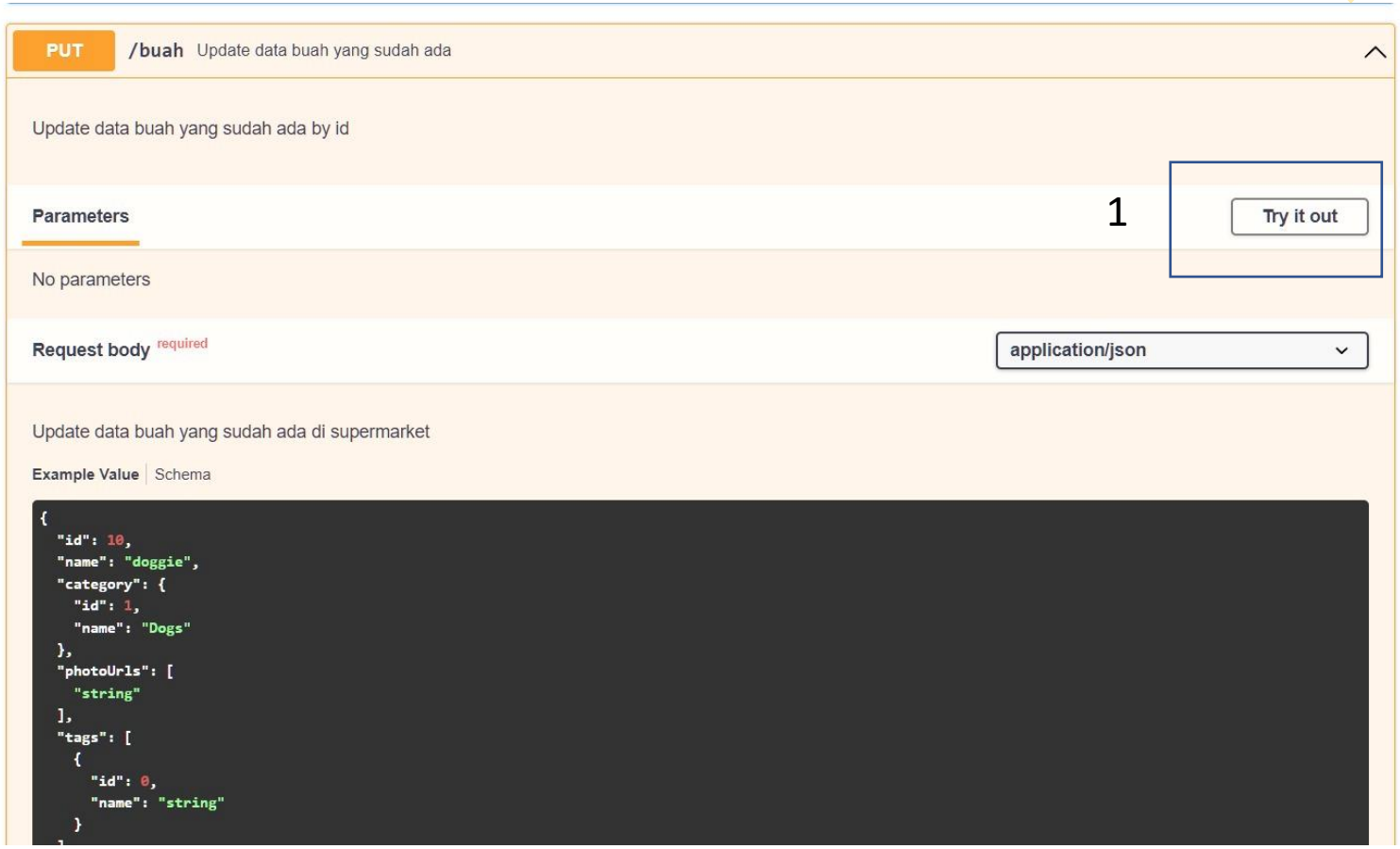




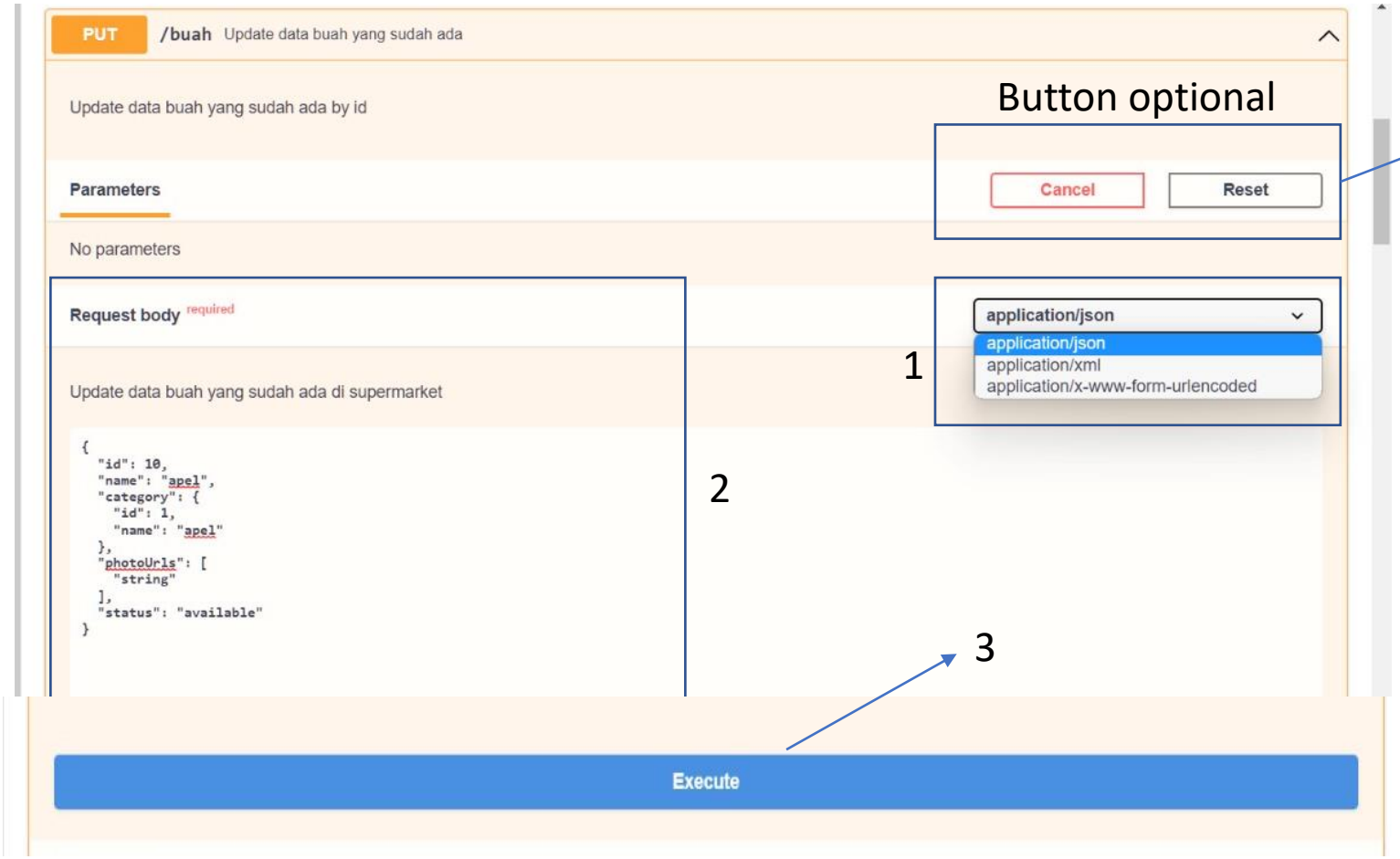
Step by Step Test API dengan Swagger

Disini saya akan menjelaskan cara test api dengan swagger menggunakan API dummy yang sudah disiapkan oleh swagger.

Pertama kita pilih dulu method yang ingin kita test, disini kita akan test method PUT, klik button **Try it Out** untuk menjalankan method yang ingin kita test



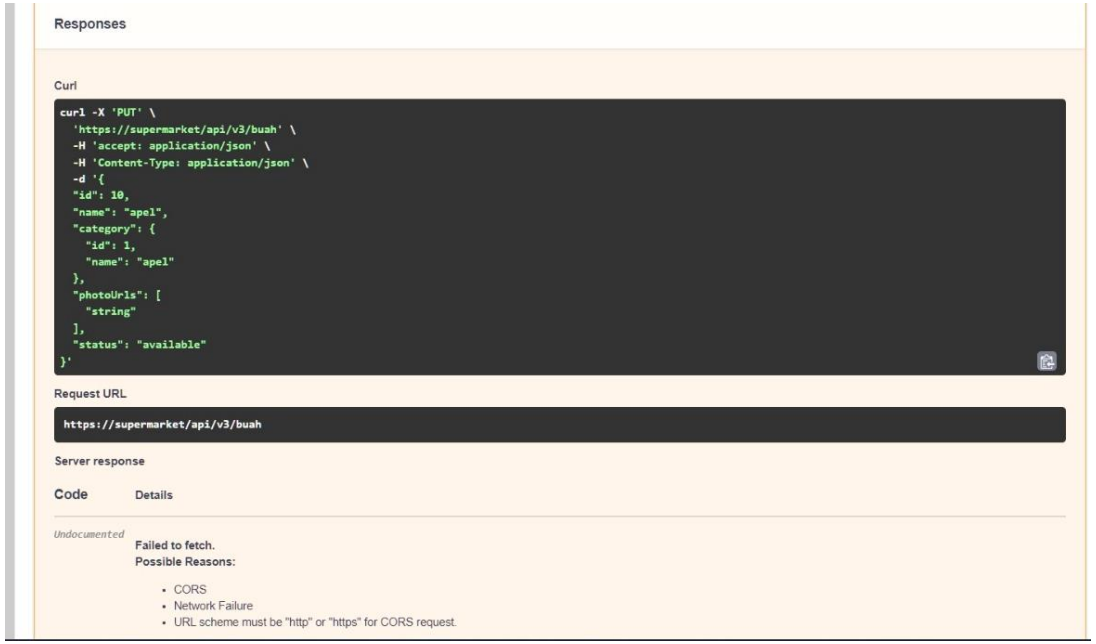
Setelah itu Langkah pertama kita bisa menentukan content/bentuk data yg mau kita coba apa misal application/json tapi secara default sudah terisi application/json. Pada kolom request body biasanya sudah disediakan requestnya jadi kita tinggal tentukan saja field mana yang mau kita update valuenya setelah itu klik tombol execute

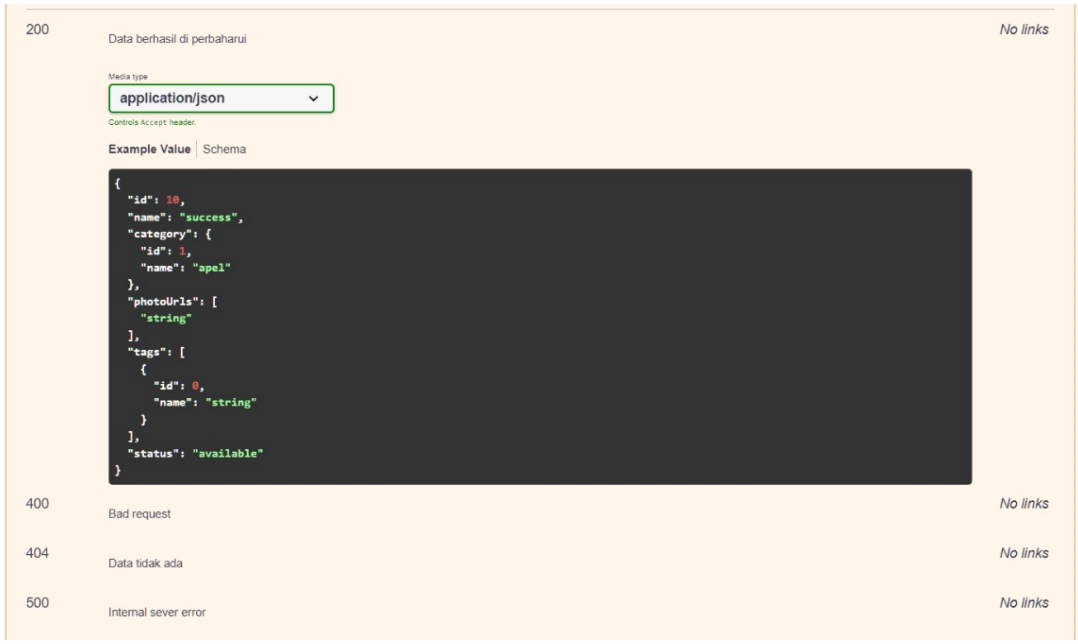


Button cancel digunakan untuk membatalkan test API dan button Reset untuk mengatur ulang request body kebentuk semula

Setelah di execute swagger akan mengeluarkan response yang terdiri dari curl,request curl, server response, http status code serta schemanya.

Pada contoh dibawah ini response yang keluar adalah 200 yang artinya data berhasil diupdate





Conclussion

- Student mengetahui apa itu swagger
- Student mengenal tools/jenis swagger
- Student mengenal fungsi swagger
- Student mengenal apa itu Api
- Student mengenal jenis-jenis Api
- Student mengenal arsitektur Api
- Student mengenal komponen swagger
- Student mengenal komponen pada object paths
- Student mengetahui step by step pembuatan dokumentasi Api
- Student mengetahui cara test Api pada swagger

Source

https://editor.swagger.io/

https://spec.openapis.org/oas/v3.1.0#response-object

https://spec.openapis.org/oas/v3.1.0#pathItemObject

https://spec.openapis.org/oas/v3.1.0#operationObject

https://spec.openapis.org/oas/v3.1.0#componentsObject

https://spec.openapis.org/oas/v3.1.0#requestBodyObject

https://rinaldiudrasuhanda.medium.com/apa-itu-swagger-a9c8157ed725

https://rinaldiudrasuhanda.medium.com/apa-itu-open-api-1b2cd2adac20