

## **BAB III PELAKSANAAN KERJA PROFESI**

### **3.1 Bidang Kerja**

Dalam melaksanakan kerja profesi Pratikan dalam hal ini berkesempatan menjadi bagian dari tim *Back-End Developer* di PT. Dengan Ide Cipta Solusi yang bertugas untuk membuat, merancang dan melakukan pengujian secara manual dan otomatis melalui *API (Application Programming Interface)* serta berkolaborasi dengan bagian *Front-End* sesuai kebutuhan aplikasi. Selama pengerjaan project untuk PT. Dengan Ide Cipta Solusi pada proses pembuatannya pratikan menggunakan beberapa aplikasi untuk bertanggung jawab melakukan pengujian sistem *API (Application Programming Interface)* dengan menggunakan (*Swagger-Ui API*), pengembangan sistem *Back-End (Java Script)* serta penyimpanan dan dokumentasi project (*Github*).

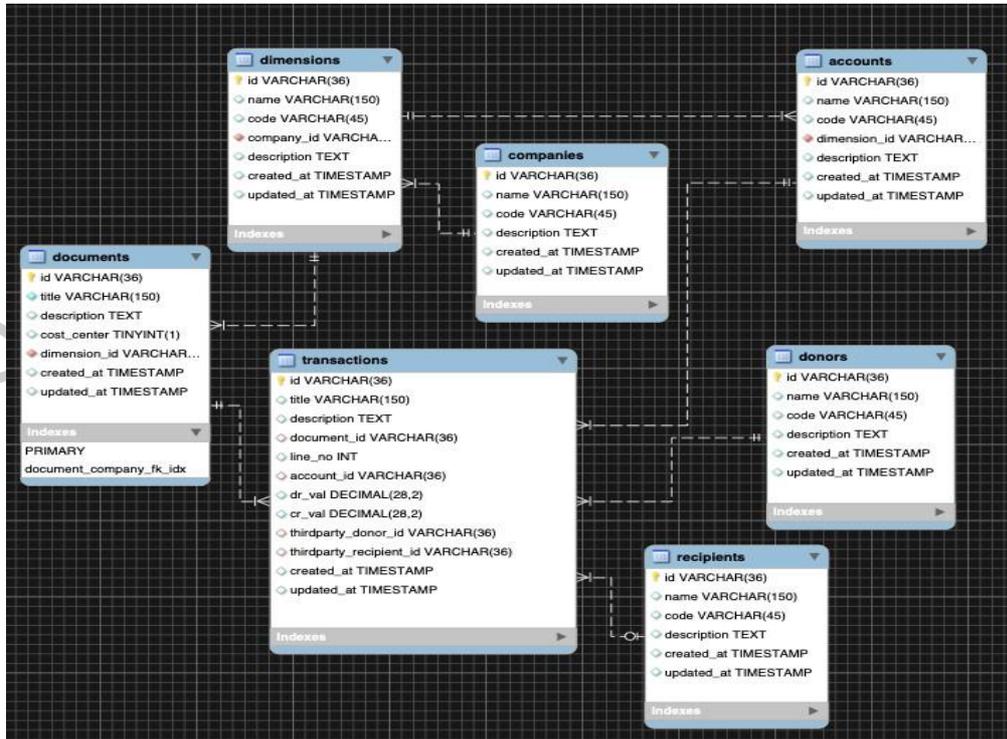
Pembuatan sistem akuntansi keuangan pada organisasi nirlaba ini mengambil referensi berdasarkan data dan kebutuhan klien untuk mendapatkan suatu aplikasi akuntansi keuangan untuk pencatatan dan pemasukan serta pengeluaran untuk organisasi nilaba. Dalam hal ini Pratikan merancang struktur database website Akuntansi yang bertujuan untuk memfasilitas organisasi nirlaba dalam melakukan pencatatan keuangan secara terkomputerisasi.

### **3.2 Pelaksanaan Kerja**

Tahap selanjutnya, seluruh tim proyek ERP4NGO melakukan pertemuan melalui *online meeting via google meet* dan bertemu untuk membahas tentang kebutuhan *project*, dan juga proses bisnisnya. Dalam perencanaan kebutuhan tim membahas tentang penggunaan tools atau Bahasa pemrograman apa saja yang nantinya akan digunakan dalam implementasi aplikasi, yakni seperti *MySQL WorkBench* untuk pembuatan

Database Website, Github untuk penyimpanan dan dokumentasi, Quarkus untuk pembuatan Rest API sekaligus melakukan testing API tersebut.

### 3.2.1 Class Diagram



Gambar 3.1 Class Diagram

Class Diagram pada gambar 3.1 diatas ini, dibutuhkan untuk menghasilkan suatu model bisnis yang akan digunakan untuk membangun suatu sistem, ERP4NGO terdapat 7 table yang akan menyimpan data-data yang nantinya akan dibutuhkan dalam pembuatan aplikasi. Berikut table tersebut : *Donor*, *transactions*, *accounts*, *companies*, *dimensions*, *documents*, dan *recipients*. Relasi antar table ditunjukkan dengan adanya kebutuhan yang akan terjadi pada table yang berhubungan.

### 3.2.2 Open Quarkus

*Rest API bisa disebut sebagai pengujian yang dilakukan untuk menemukan bug dan error selama proses pengujian sistem website dengan framework yang diberikan. Pengujian*

sering dilakukan karena waktu yang diperlukan untuk menyelesaikan pengujian sangat singkat dan kesalahan pengujian situs dapat diminimalisir. Salah satu framework yang digunakan praktisi untuk menguji REST API adalah QuarkusAPI. Dalam hal ini QuarkusAPI merupakan aplikasi yang berperan sebagai REST client untuk REST API ERP4NGO.

### 3.2.3 API (Application Programming Interface)

API (Application Programming Interface) merupakan interface yang dapat menghubungkan *client* dengan *server*. API sangat penting dalam sebuah pembuatan website ERP4NGO karena dengan adanya API tersebut dapat menghubungkan dua komponen aplikasi yang saling berinteraksi melalui serangkaian protokol pada satu *platform* yang sama. API bekerja dengan cara *user* mengakses aplikasi, lalu API akan membuat permintaan ke *server*, lalu *server* me-*respons* API. Terdapat beberapa *method* yang digunakan ERP4NGO sebagai berikut.

Method	Fungsi
GET	Menampilkan data yang di <i>request client</i>
POST	Create data
PUT	Melakukan edit dan <i>update</i> ke seluruh data, sesuai ID yang dipilih
DELETE	Menghapus data yang dipilih IDnya

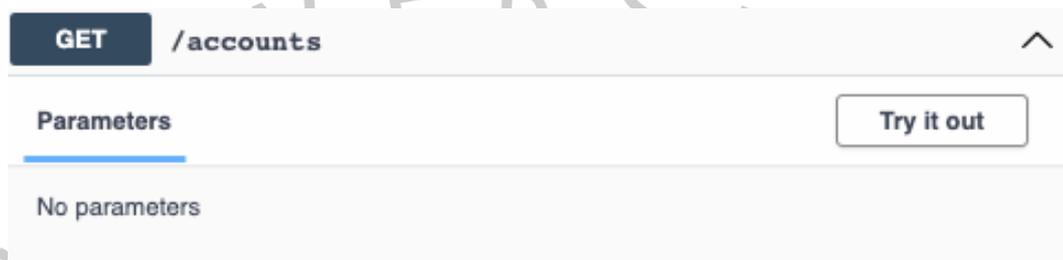
Tabel 3.11 Fungsi Method REST API

### 3.2.4 Resource Account

### 1. Method GET

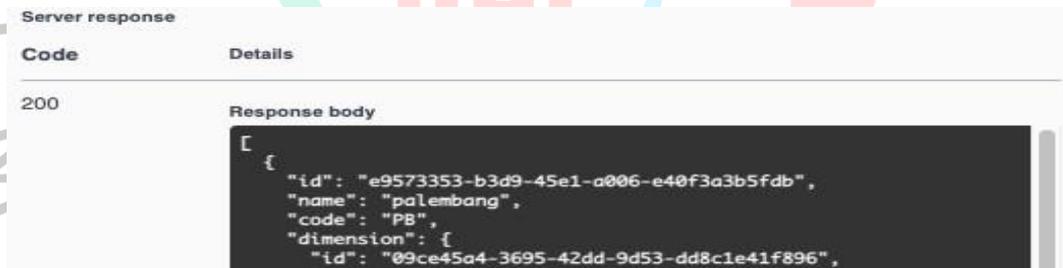
Pengujian API yang dilakukan untuk *moduls Account*, yang berfungsi menampilkan data seperti nama, *code*, *description* kemudian data tersebut akan ditampung oleh *server*. Berikut cara kerja pungujian API melalui *Quarkus*.

- Pilih *method GET*.



Gambar 3.2 Method GET API

- Lalu klik *button try it out*, seperti gambar 3.2 diatas.
- Setelah itu Quarkus memberikan *respons*-nya seperti gambar 3.3 berikut ini.



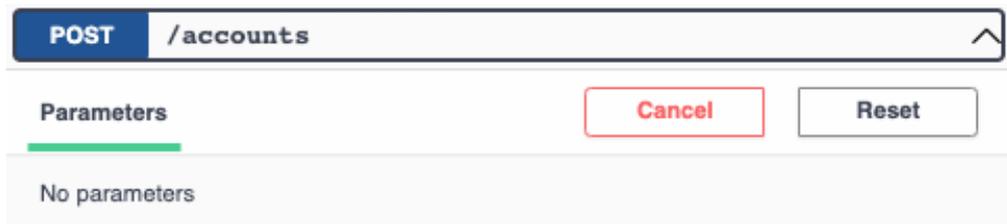
Gambar 3.3 Hasil Method GET

### 2. Method POST

Pengujian API yang dilakukan untuk *method Account*, yang berfungsi menampilkan data dan menarik data *dimension* melalui *ID*. Berikut data ketentuannya seperti nama, *code*, *description* dan *ID dimension*. Berikut cara kerja pungujian API melalui *Quarkus*.

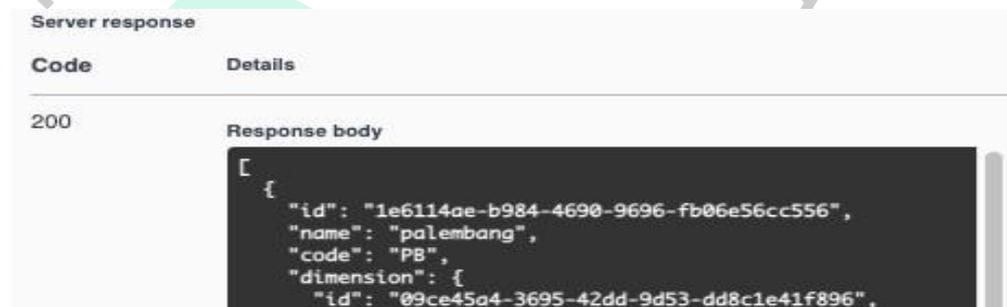
- Pilih *method POST*.
- Isi data sesuai ketentuan seperti: *title*, *description*, dan sertakan *ID dimension*, karena menarik data.

- Lalu klik *button try it out*, seperti pada gambar 3.4 berikut ini.



Gambar 3.4 Tampilan Method POST API

- Lalu *Quarkus* akan memberikan *respons*-nya, seperti gambar 3.5 dibawah ini.



Gambar 3.5 Tampilan Hasil Method POST API

### 3. Method PUT

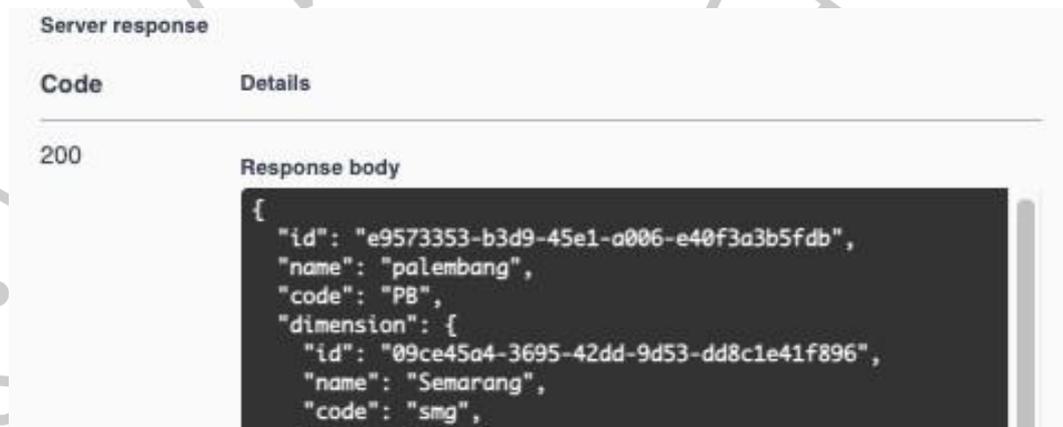
Pengujian *API* yang dilakukan untuk *method PUT Account*, dilakukan untuk mengedit suatu data yang telah di-*request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian *API* melalui *Quarkus* :

- Pilih *method PUT*.
- Ketik *ID Account* yang ingin diedit datanya, seperti gambar 3.6 berikut ini.



Gambar 3.6 Tampilan Method PUT API

- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.
- Lalu *execute* seperti gambar 3.7 dibawah ini.



Gambar 3.7 Tampilan Hasil Method PUT API

#### 4. Method DELETE

Pengujian API yang dilakukan untuk *method DELETE Account*, dilakukan untuk menghapus suatu data yang telah di-*request client* dan data yang ditarik juga akan terhapus. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method DELETE*.
- Ketik/ambil ID yang ingin di *delete*.
- Lalu *execute*.
- Lalu *Quarkus API* akan memberikan *respons*-nya, bahwa data telah terhapus. Seperti gambar 3.8 berikut ini.



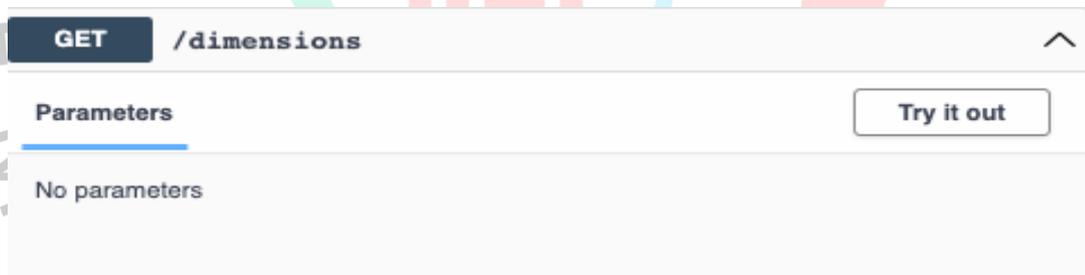
Gambar 3.8 Tampilan Method DELETE API

### 3.2.5 Resource Company

#### 1. Method GET

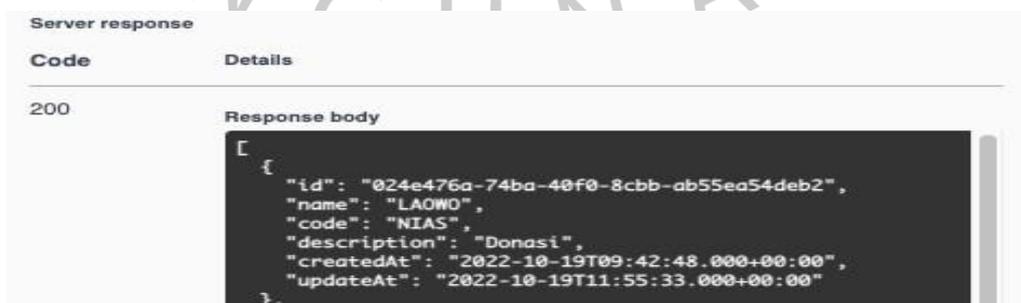
Pengujian API yang dilakukan untuk *method company*, yang berfungsi menampilkan data seperti nama, *code*, *description* kemudian data tersebut akan ditampung oleh server. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method GET*.
- Lalu klik *button try it out*, seperti gambar 3.9 berikut ini.
- Masukan *request id*.



Gambar 3.9 Tampilan Method GET API

- Setelah itu Quarkus memberikan *respons*-nya seperti gambar 3.10 berikut ini.

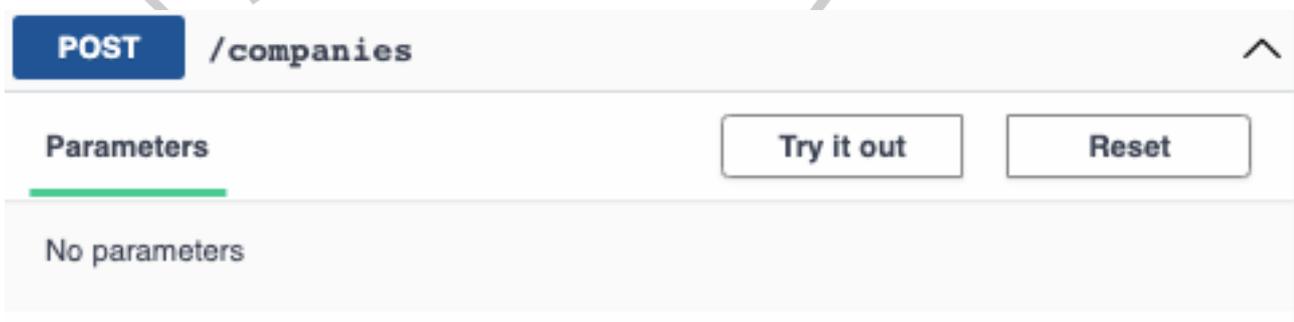


Gambar 3.10 Tampilan Hasil Method GET API

## 2. Method POST

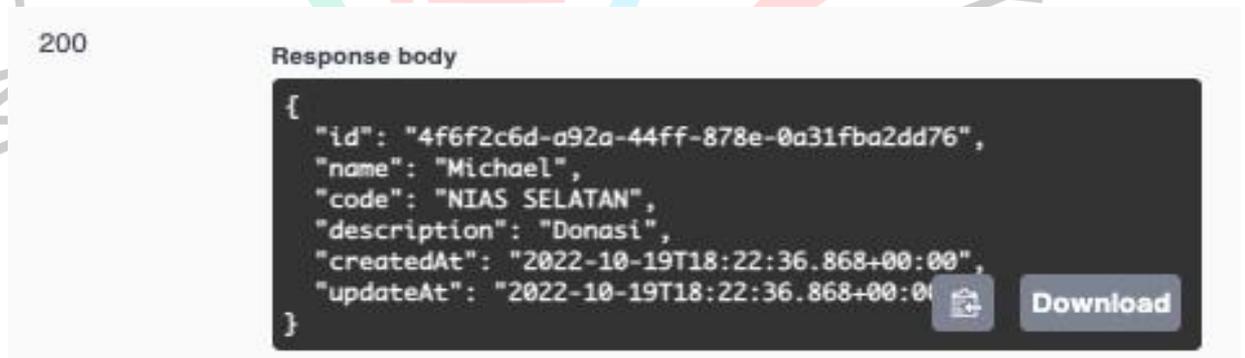
Pengujian API yang dilakukan untuk *method POST company*, yang dilakukan untuk *request* dan mengirimkan nilai data seperti nama, *code*, *description*, kemudian data tersebut dikirim ke *server* untuk ditampilkan. Berikut cara kerja pengujian API melalui *Quarkus*.

- Pilih *method POST*.
- Lalu isi data seperti, nama, *code*, *description*, dll



**Gambar 3.11 Tampilan Method POST API**

- Klik data sesuai ketentuan, lalu *Quarkus* akan memberikan *respons*-nya seperti gambar 3.12 berikut ini.



**Gambar 3.12 Tampilan Hasil Method POST API**

### 3. Method PUT

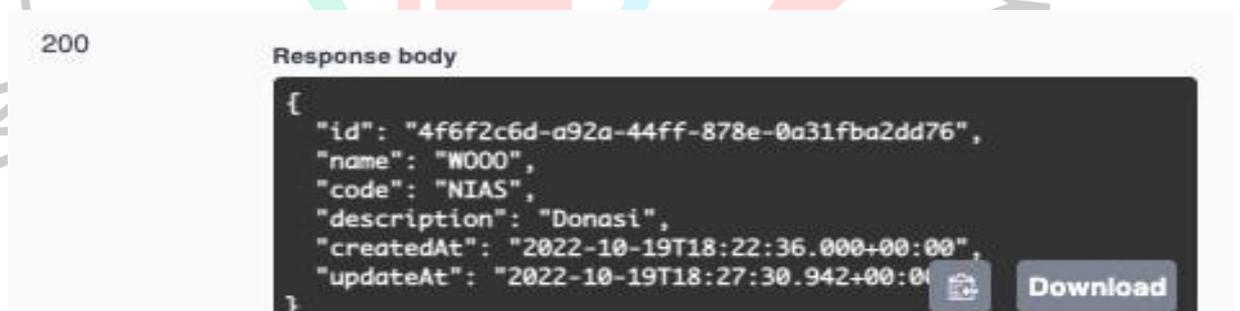
Pengujian API yang dilakukan untuk *method PUT company*, dilakukan untuk mengedit suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui *Quarkus*.

- Pilih *method PUT*.
- Tulis Id yang ingin diedit datanya.
- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.



Gambar 3.13 Tampilan Method PUT API

- Setelah itu *Quarkus* mengirim *respons*-nya, seperti gambar 3.14 berikut ini.



Gambar 3.14 Tampilan Hasil Method PUT API

#### 4. Method DELETE

Pengujian API yang dilakukan untuk method *DELETE company*, dilakukan untuk menghapus suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*.

Berikut cara kerja pengujian API melalui *Quarkus*.

- Pilih *method DELETE*.
- Ketik/ambil ID yang ingin di *delete*.
- Klik *button try it out*.
- Lalu *Quarkus API* akan memberikan *respons*-nya seperti gambar 3.15 berikut ini.



Gambar 3.15 Tampilan Method DELETE API

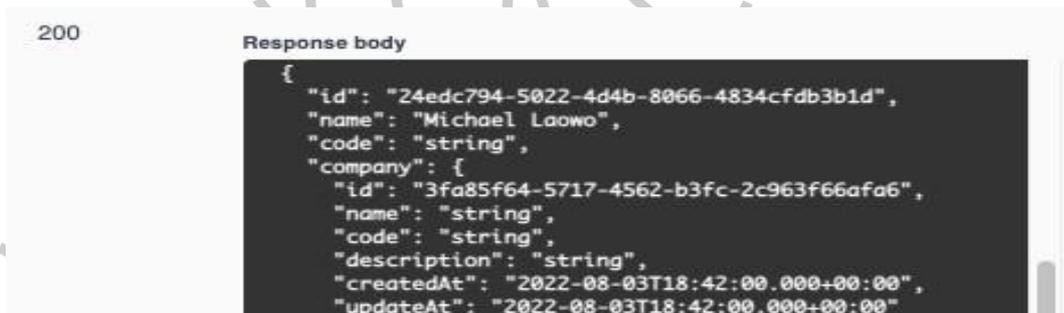
### 3.2.6 Resource Dimension

#### 1. Method GET

Pengujian API yang dilakukan untuk *method dimension*, yang berfungsi menampilkan data cabang masjid seperti nama,

*code*, *description* kemudian data tersebut akan ditampung oleh *server*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method GET*.
- *Execute*.
- Setelah itu *Quarkus API* akan menampilkan data seperti gambar 3.16 berikut ini.



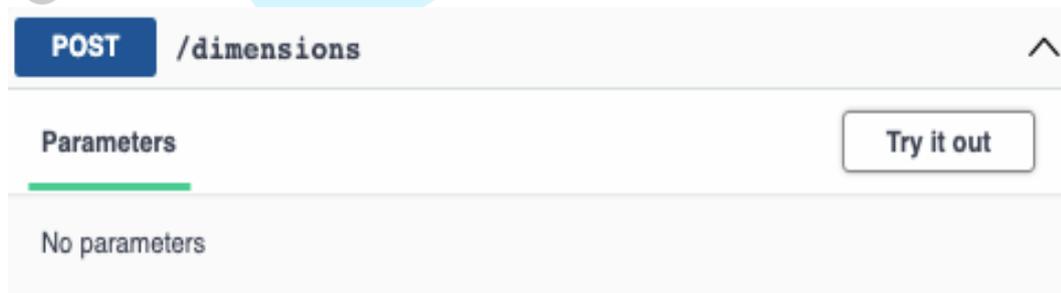
```
200
Response body
{
  "id": "24edc794-5022-4d4b-8066-4834cfdb3b1d",
  "name": "Michael Laowo",
  "code": "string",
  "company": {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "name": "string",
    "code": "string",
    "description": "string",
    "createdAt": "2022-08-03T18:42:00.000+00:00",
    "updatedAt": "2022-08-03T18:42:00.000+00:00"
  }
}
```

Gambar 3.16 Tampilan Hasil Method GET API

## 2. Method POST

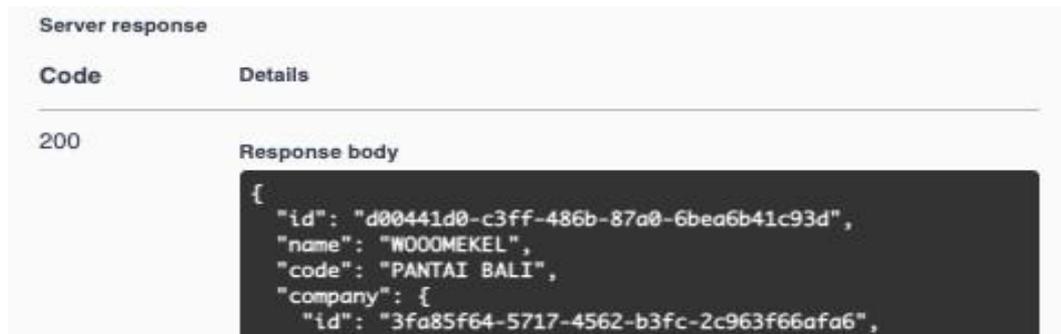
Pengujian *API* yang dilakukan untuk *method POST dimension*, yang dilakukan untuk *request* dan mengirimkan nilai data seperti nama, *code*, *description*, kemudian data tersebut dikirim ke *server* untuk ditampung. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method POST*.
- Lalu isi data seperti, nama, *code*, *description*, dll.



Gambar 3.17 Tampilan Method POST API

- Kemudian *execute* dan *Quarkus* akan memberikan *respons*-nya, seperti gambar 3.18 berikut ini.

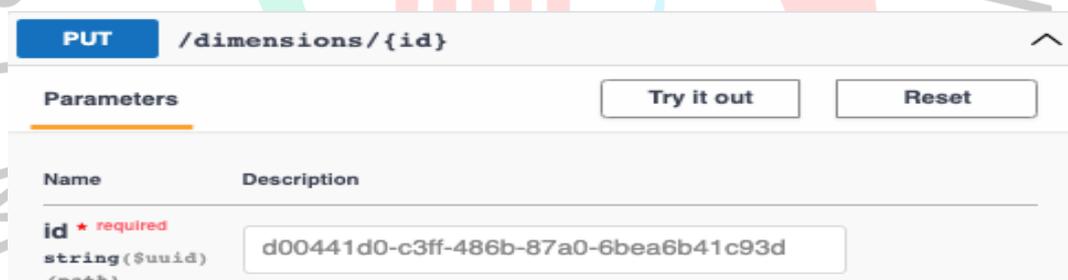


Gambar 3.18 Tampilan Hasil Method POST API

### 3. Method PUT

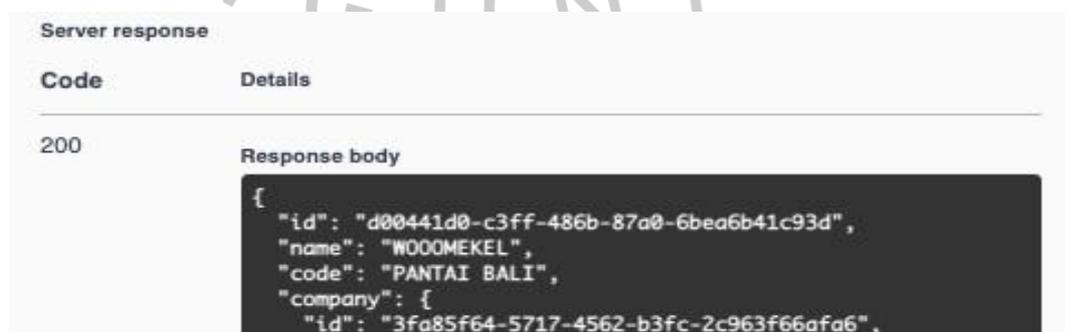
Pengujian API yang dilakukan untuk *method PUT dimension*, dilakukan untuk mengedit suatu data yang telah di-request *client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method PUT*.
- Tulis *ID* yang ingin diedit datanya seperti gambar 3.19.



Gambar 3.19 Tampilan Method PUT API

- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.
- Lalu klik *button try it out*.
- Setelah itu Quarkus mengirim *respons*-nya, seperti gambar 3.20 berikut ini.



Gambar 3.20 Tampilan Hasil Method POST API

#### 4. Method DELETE

Pengujian API yang dilakukan untuk *method DELETE dimension*, dilakukan untuk menghapus suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui *Quarkus*.

- Pilih *method DELETE*.
- Ketik/ambil *ID* yang ingin di *delete*.
- *Execute*.
- Lalu *Quarkus API* akan memberikan *respons*-nya seperti gambar 3.21 berikut ini.



Gambar 3.21 Tampilan Method DELETE API

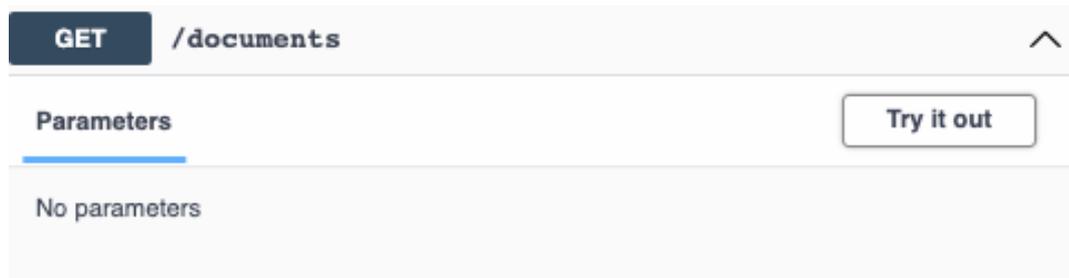
### 3.2.7 Resource Document

#### 1. Method GET

Pengujian API yang dilakukan untuk *method document*, yang berfungsi menampilkan data seperti nama, *code*, *description* kemudian data tersebut akan ditampung oleh *server*. Berikut cara kerja pengujian API melalui *Quarkus*.

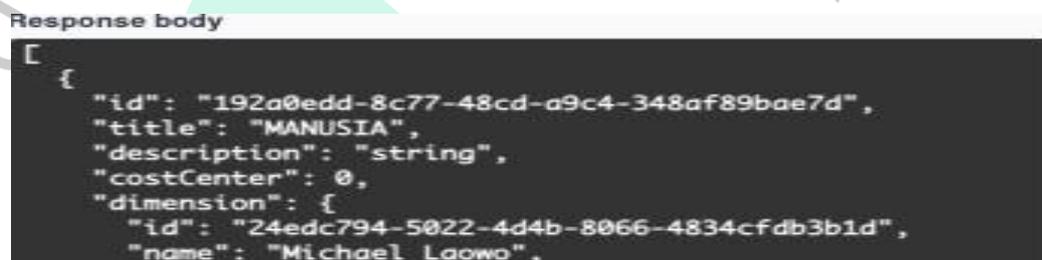
- Pilih *method GET*.
- Masukkan *request id* sesuai ketentuan.

- Klik *button try it out* seperti gambar 3.22 berikut ini.



Gambar 3.22 Tampilan Method GET API

- Setelah itu *Quarkus API* akan menampilkan data seperti gambar 3.23 berikut ini.

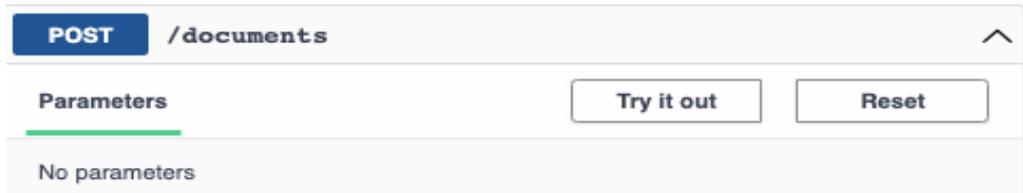


Gambar 3.23 Tampilan Hasil Method GET API

## 2. Method POST

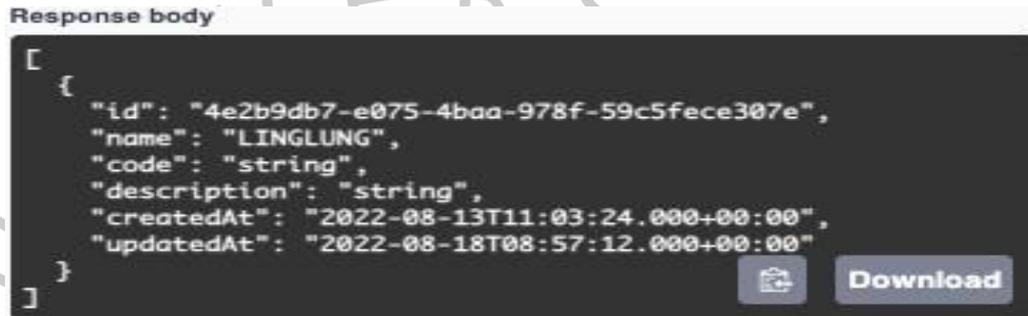
Pengujian *API* yang dilakukan untuk *method POST document*, yang dilakukan untuk request dan mengirimkan nilai data seperti nama, *code*, *description*, kemudian data tersebut dikirim ke *server* untuk ditampung. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method POST*.
- Lalu isi data seperti, nama, *code*, *description*, dll
- Klik *button try it out* seperti gambar 3.24 berikut ini.



Gambar 3.24 Tampilan Method POST API

- Kemudian *Quarkus* akan memberikan *respons*-nya, seperti gambar 3.25 berikut ini.

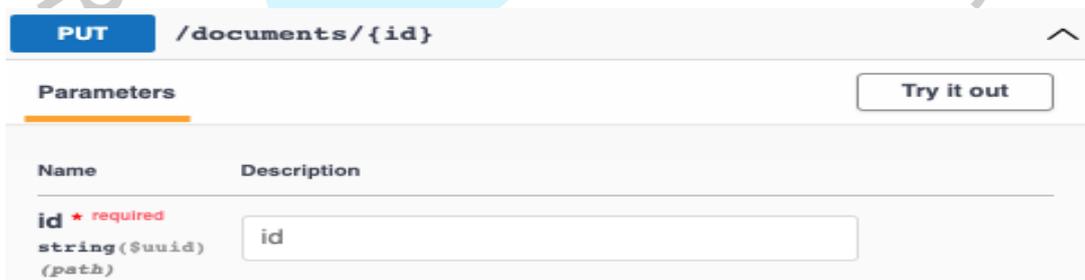


Gambar 3.25 Tampilan Hasil Method POST API

### 3. Method PUT

Pengujian *API* yang dilakukan untuk *method PUT document*, dilakukan untuk mengedit suatu data yang telah di-*request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method PUT*.
- Tulis *Id* yang ingin diedit datanya seperti gambar 3.26 ini.



Gambar 3.26 Tampilan Method PUT API

- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.

- *Execute* seperti gambar 3.27 berikut ini.

Response body

```
{
  "id": "4e2b9db7-e075-4baa-978f-59c5fece307e",
  "name": "MEKEL",
  "code": "OTW BALI",
  "description": "PANTAI BALI",
  "createdAt": "2022-08-13T11:03:24.000+00:00",
  "updatedAt": "2022-10-20T16:41:03.269+00:00"
}
```

Download

Gambar 3.27 Tampilan Hasil Method PUT API

#### 4. Method DELETE

Pengujian API yang dilakukan untuk *method DELETE document*, dilakukan untuk menghapus suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui Quarkus.

### 3.2.8 Resource Donor

#### 1. Method GET

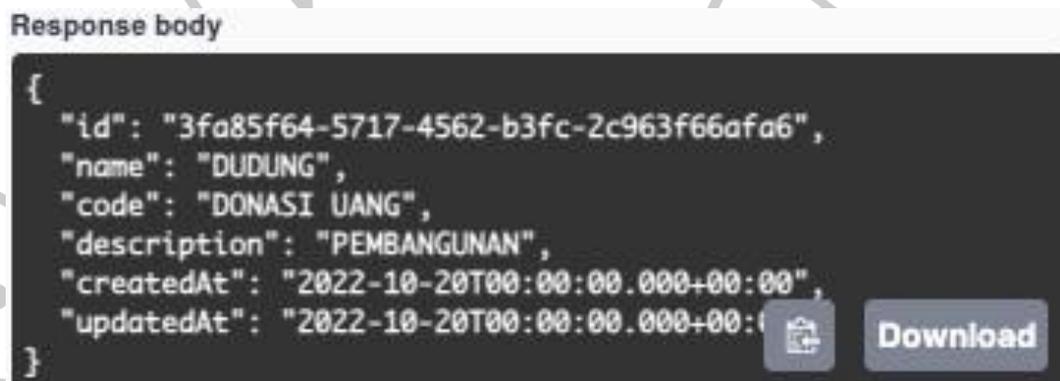
Pengujian API yang dilakukan untuk *method donor*, yang berfungsi menampilkan data seperti nama, *code*, *description* kemudian data tersebut akan ditampung oleh *server*. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method GET*.
- Masukkan *request id* sesuai ketentuan.
- Klik *button try it out* seperti gambar 3.28 berikut ini.



Gambar 3.28 Tampilan Method GET API

- Lalu *Quarkus* akan menampilkan *request* datanya, seperti gambar 3.29 berikut ini.

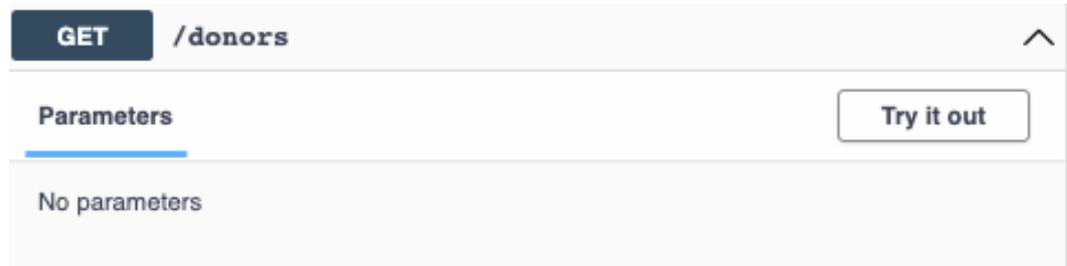


Gambar 3.29 Tampilan Hasil Method GET API

## 2. Method POST

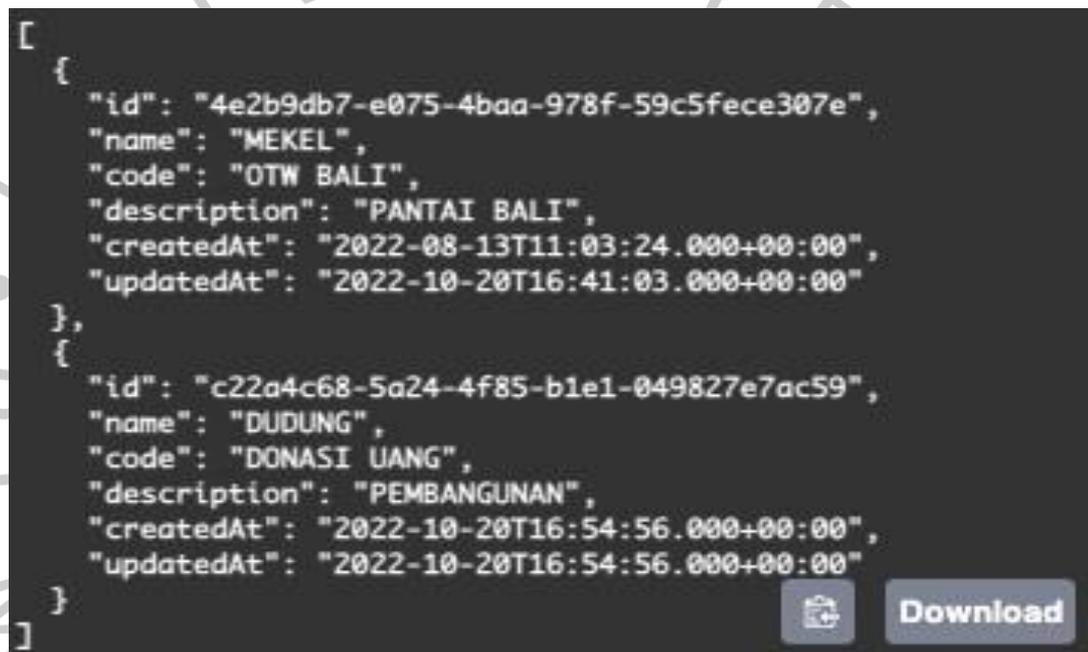
Pengujian *API* yang dilakukan untuk *method POST donor*, yang dilakukan untuk request dan mengirimkan nilai data seperti nama, *code*, *description*, kemudian data tersebut dikirim ke *server* untuk ditampung. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method POST*.
- Lalu isi data seperti, nama, *code*, *description*, dll
- Klik *button try it out* seperti gambar 3.30 berikut ini.



Gambar 3.30 Tampilan Method POST API

- Lalu *Quarkus API* akan memberikan *respons*-nya, seperti gambar 3.31 berikut ini.

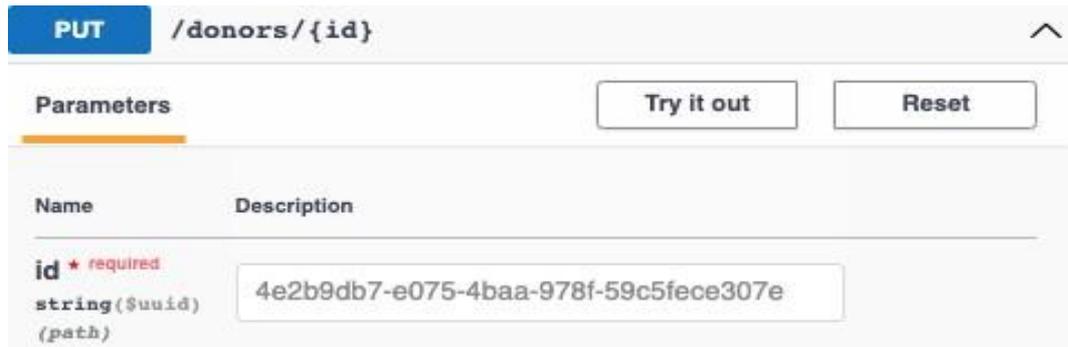


Gambar 3.31 Tampilan Hasil Method POST API

### 3. Method PUT

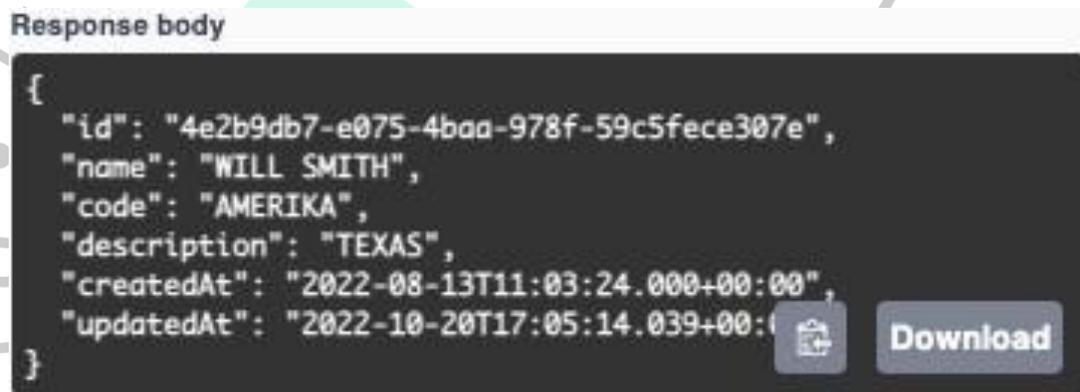
Pengujian *API* yang dilakukan untuk *method PUT donor*, dilakukan untuk mengedit suatu data yang telah di-*request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method PUT*.
- Tulis *ID* yang ingin diedit.
- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.



Gambar 3.32 Tampilan Method PUT API

- Lalu klik *button try it out* seperti gambar 3.32 diatas ini.
- Setelah itu *Quarkus* mengirim *respons*-nya, seperti gambar 3.33 berikut dibawah ini.

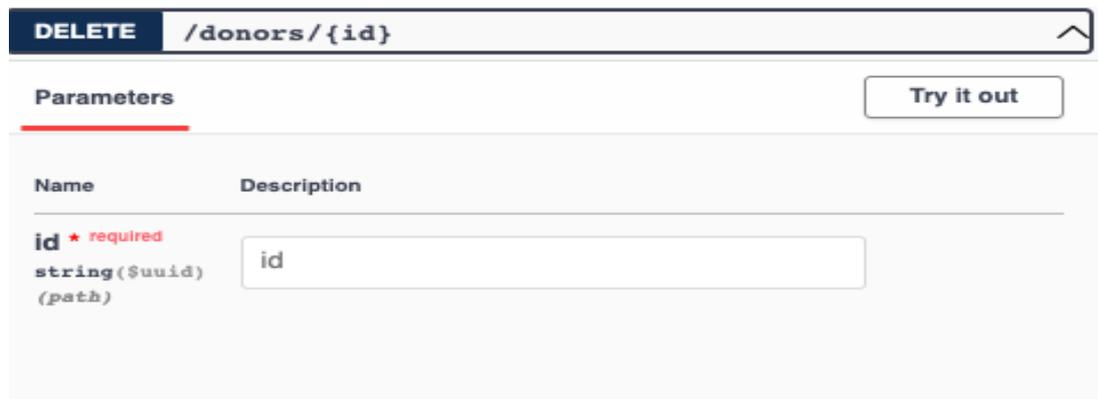


Gambar 3.33 Tampilan Hasil Method PUT API

#### 4. Method DELETE

Pengujian *API* yang dilakukan untuk *method DELETE* donor, dilakukan untuk menghapus suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method DELETE*.
- Ketik/ambil *ID* yang ingin di delete.
- *Execute*.
- Lalu *Quarkus API* akan memberikan *respons*-nya seperti gambar 3.34 berikut ini.



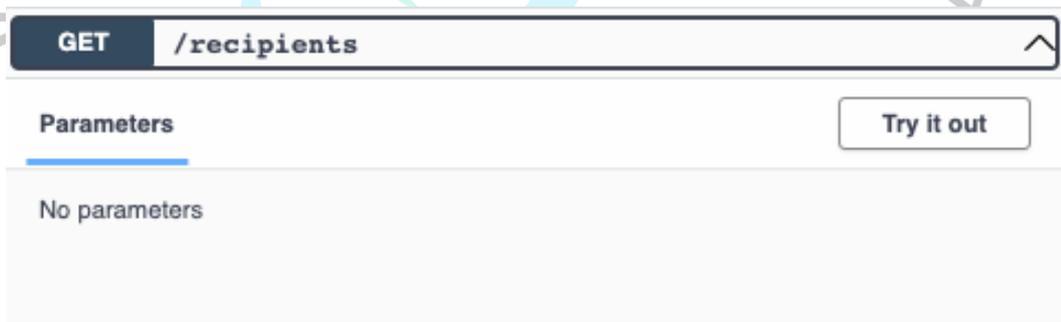
Gambar 3.34 Tampilan Method DELETE API

### 3.2.9 Resource Recipient

#### 1. Method GET

Pengujian *API* yang dilakukan untuk *method Recipient*, yang berfungsi menampilkan data seperti nama, *code*, *description* kemudian data tersebut akan ditampung oleh *server*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method GET*.
- Masukkan *request id* sesuai ketentuan.
- Klik *button try it out* seperti gambar 3.35 berikut ini.



Gambar 3.35 Tampilan Method GET API

- *Quarkus* akan menampilkan *respons request* datanya, seperti gambar 3.36 berikut ini.

```
Response body
{
  "id": "6d1763f0-a155-4ae9-a285-9367bf35ada1",
  "name": "OKEEY",
  "code": "string",
  "description": "string",
  "createdAt": "2022-08-13T09:28:13.000+00:00",
  "updatedAt": "2022-08-25T14:36:25.000+00:00"
},
{
  "id": "8ebf1c93-9d1a-4182-ac4f-2682d1c5cc78",
  "name": "PUBG",
  "code": "chiken",
  "description": "string",
  "createdAt": "2022-08-25T12:57:30.000+00:00",
  "updatedAt": "2022-08-25T12:57:30.000+00:00"
},
{
  "id": "d96c523c-b382-4b76-9860-80a82f915589",
  "name": "W001810",
  "code": "004",
  "description": "DONASI GERBANG",
  "createdAt": "2022-10-20T17:24:22.000+00:00",
  "updatedAt": "2022-10-20T17:24:22.000+00:00"
}
}
```

Gambar 3.36 Tampilan Hasil Method GET API

## 2. Method POST

Pengujian API yang dilakukan untuk *method POST Recipient*, yang dilakukan untuk request dan mengirimkan nilai data seperti nama, *code*, *description*, kemudian data tersebut dikirim ke server untuk ditampilkan. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method POST*.
- Lalu isi data seperti, nama, *code*, *description*, dll
- Klik *button try it out* seperti gambar 3.37 berikut ini.



Gambar 3.37 Tampilan Method POST API

- Kemudian *execute* dan *Quarkus* akan memberikan *respons*-nya, seperti gambar 3.38 berikut ini.

Response body

```
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "name": "W001810",
  "code": "004",
  "description": "DONASI GERBANG",
  "createdAt": "2022-10-20T00:00:00.000+00:00",
  "updatedAt": "2022-10-20T00:00:00.000+00:00"
}
```

Gambar 3.38 Tampilan Hasil Method POST API

### 3. Method PUT

Pengujian API yang dilakukan untuk *method PUT Recipient*, dilakukan untuk mengedit suatu data yang telah di-request *client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui *Quarkus*.

- Pilih *method POST*.
- Tulis *ID* yang ingin diedit.
- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.
- Klik *button try it out* seperti gambar 3.39 berikut ini.



Gambar 3.39 Tampilan Method PUT API

- Setelah itu *Quarkus* mengirim *respons*-nya, seperti gambar 3.40 berikut ini.

Response body

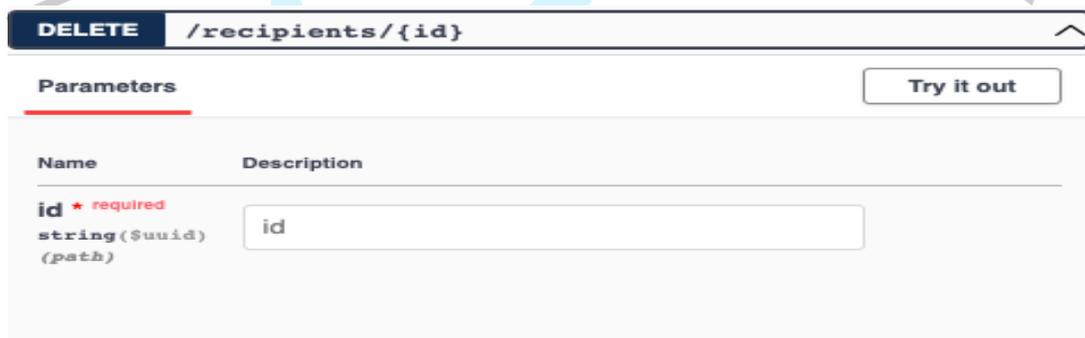
```
{
  "id": "d96c523c-b382-4b76-9860-80a82f915589",
  "name": "BOKIS",
  "code": "005",
  "description": "DONASI NASI KOTAK",
  "createdAt": "2022-10-20T17:24:22.000+00:00",
  "updatedAt": "2022-10-20T17:38:58.984+00:00"
}
```

Gambar 3.40 Tampilan Hasil Method PUT API

#### 4. Method DELETE

Pengujian API yang dilakukan untuk *method DELETE Recipient*, dilakukan untuk menghapus suatu data yang telah di *request client*. Data tersebut seperti nama, *code*, *description*. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method DELETE*.
- Ketik/ambil *ID* yang ingin di *delete*.
- *Execute*.
- Lalu *Quarkus API* akan memberikan *respons-nya* seperti gambar 3.41 berikut ini.



Gambar 3.41 Tampilan Method DELETE API

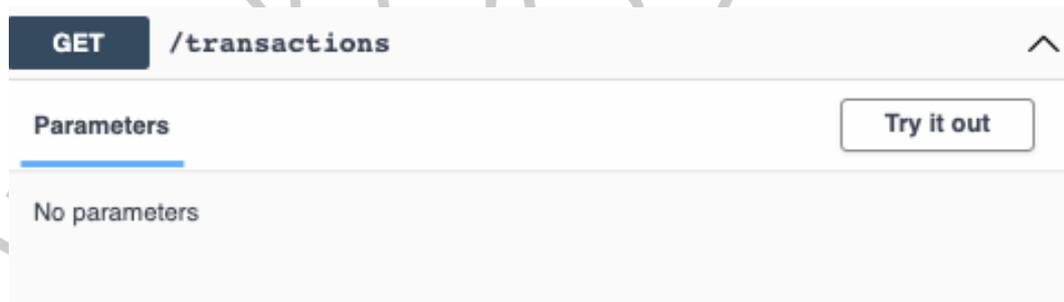
### 3.2.10 Resource Transaction

#### 1. Method GET

Pengujian API yang dilakukan untuk *method transaction*, yang berfungsi menampilkan data seperti nama, *code*, *title*,

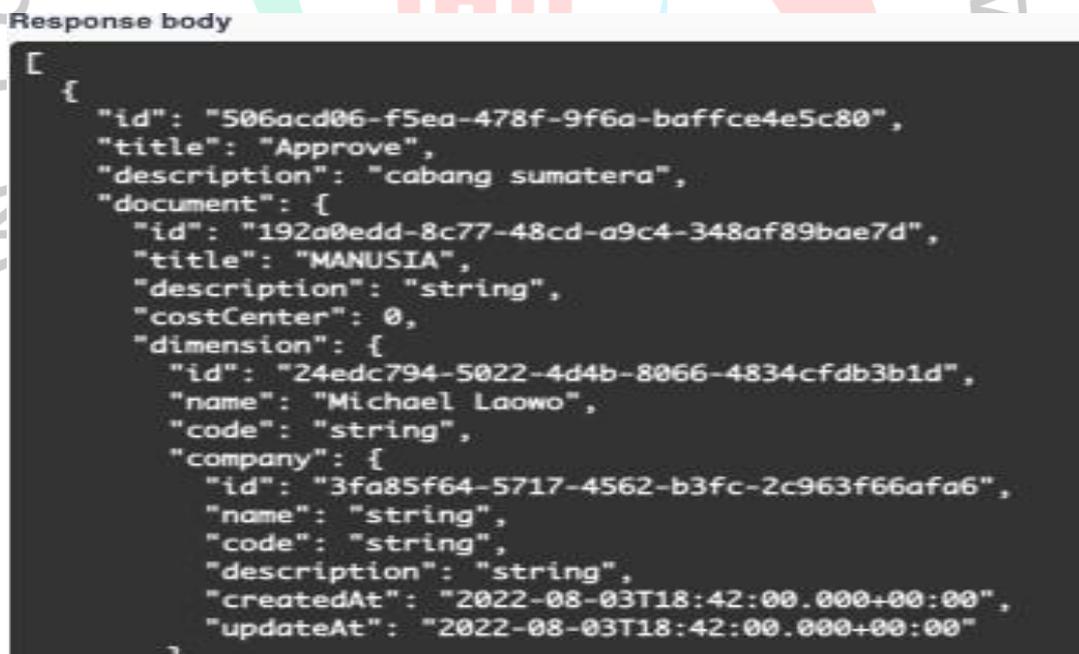
*description, document, line.no, id account, donor id, recipient id.* kemudian data tersebut akan ditampung oleh server. Berikut cara kerja pengujian API melalui Quarkus.

- Pilih *method GET*.
- Masukkan *request id* sesuai ketentuan.
- Klik *button try it out* seperti gambar 3.42 berikut ini.



Gambar 3.42 Tampilan Method GET API

- Lalu Quarkus akan menampilkan *request* datanya, seperti gambar 3.43 berikut ini.



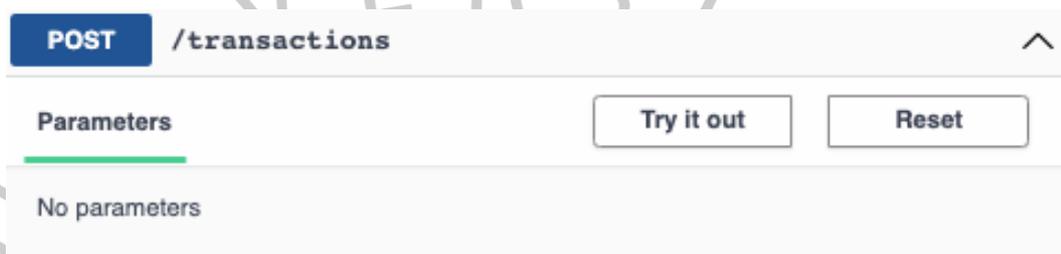
Gambar 3.43 Tampilan Hasil Method GET API

## 2. Method POST

Pengujian API yang dilakukan untuk *method POST Transaction*, yang dilakukan untuk *request* dan mengirimkan nilai data seperti nama, *code, title, description document*,

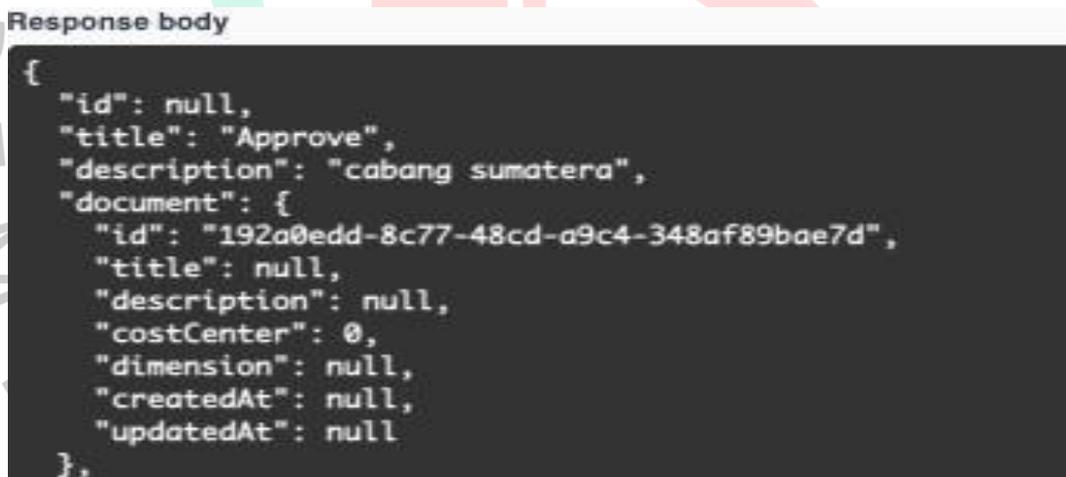
*line.no, id account, donor id, recipient id.* kemudian data tersebut dikirim ke *server* untuk ditampung. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method POST*.
- Lalu isi data seperti, nama, code, description, dll
- Klik *button try it out* seperti gambar 3.44 berikut ini.



**Gambar 3.44 Tampilan Method POST API**

- Kemudian *execute* dan *Quarkus API* akan memberikan *respons*-nya, seperti gambar 3.45 berikut ini.



**Gambar 3.45 Tampilan Hasil Method POST API**

### 3. *Method PUT*

Pengujian *API* yang dilakukan untuk *method PUT Transaction*, yang dilakukan untuk *request* dan mengirimkan nilai data seperti *id, nama, code, title, description id\_document, costCenter, id account, dimension id.* kemudian data tersebut dikirim ke *server* untuk ditampung. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method PUT*.

- Tulis *ID* yang ingin diedit.
- Isi data seperti nama, *code*, *descriptions* yang ingin dirubah.
- Klik *button try it out* seperti gambar 3.46 berikut ini.

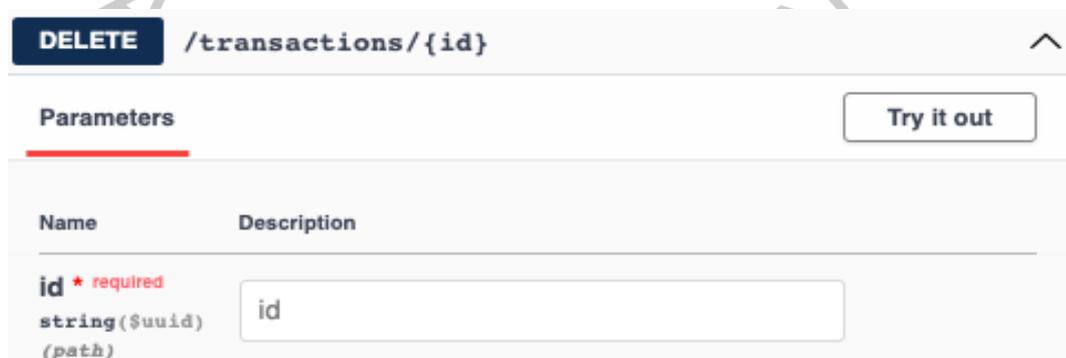


Gambar 3.46 Tampilan Method PUT API

#### 4. Method DELETE

Pengujian *API* yang dilakukan untuk *method DELETE Transaction*, dilakukan untuk menghapus suatu data yang telah di *request client*. Berikut cara kerja pengujian *API* melalui *Quarkus*.

- Pilih *method DELETE*.
- Ketik/ambil *ID* yang ingin di *delete*.
- *Execute*.
- Lalu *Quarkus API* akan memberikan *respons*-nya seperti gambar 3.47 berikut ini.



Gambar 3.47 Tampilan Method DELETE API

### 3.3 Kendala Yang Dihadapi

Terdapat beberapa kendala yang dihadapi praktikan selama masa Kerja Profesi berlangsung. Beberapa kendala yang dihadapi yaitu Pada pelaksanaan Kerja Profesi (KP) disaat pandemi Covid-19,

1. Pratkan kesulitan untuk berkomunikasi dan mengadakan pertemuan langsung dengan tim pengembangan TI. Hal ini menyebabkan kesulitan dalam berdiskusi dan mencari informasi tentang aplikasi yang akan kita kembangkan.
2. Sering juga terjadi kesalahpahaman komunikasi antara Pratkan dengan tim pengembang TI.
3. Pratkan diawal pertemuan tidak mengetahui cara pengujian API (Application Programming Interface) otomatis dengan framework Quarkus di awal pekerjaannya, sehingga harus mempelajarinya terlebih dahulu.

### 3.4 Cara Mengatasi Kendala

Cara Pratkan Mengatasi berbagai kendala dengan mencari referensi dari banyak sumber melalui internet terkait cara mengoperasikan Rest API Quarkus, melakukan pertemuan secara online dan rutin dengan bantuan aplikasi Google Meet dan berkomunikasi dengan Telegram dengan seluruh Tim dan berdiskusi bersama mengenai kebutuhan aplikasi.

### 3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

1. Praktikan mendapatkan pembelajaran mengenai cara bekerjasama yang baik dengan tim dan manajemen waktu dalam menghasilkan aplikasi yang dibutuhkan klien.
2. Praktikan mendapatkan pengetahuan yang lebih dalam mengenai perancangan Aplikasi *Back-End Developer* yang baik untuk suatu aplikasi, yakni ERP4NGO.
3. Mahasiswa mampu beradaptasi dengan lingkungan kerja dan berkolaborasi bersama *team developers*