

BAB III METODE PENELITIAN

3.1 Instrumen Penelitian

Penyusunan suatu proyek penelitian, sesuai dengan data-data permasalahan pokok yang dihadapi. Data dapat dikatakan baik apabila data tersebut dapat mewakili objek yang sedang diteliti dan untuk mendapatkan data yang baik diperlukan metode atau serangkaian cara yang sesuai dengan kebutuhan penelitian.

3.2 Prosedur Pengambilan Data

Untuk mendapatkan data yang benar, akurat dan relevan serta sesuai dengan sumber data dan tujuan penyusunan Laporan Kuliah Kerja Industri ini, maka penulis dalam pengumpulan data menggunakan beberapa teknik antara lain adalah sebagai berikut :

3.2.1 Observasi

Teknik pengumpulan data dengan cara melakukan pengamatan dan terlibat secara langsung di dalam sistem pengembangan yang telah ada di dalam Tim Pengembang Aplikasi Tukutu yang berada di bawah pengawasan PT Dinustek.

3.2.2 Studi Literatur

Melakukan berbagai pencarian terhadap teori-teori dan literatur yang dapat digunakan sebagai dasar melakukan penelitian. Serta mempelajari penelitian yang pernah dilakukan sebelumnya sehingga dapat lebih memahami teori yang ada.

3.2.3 Forum Diskusi dan Tutorial

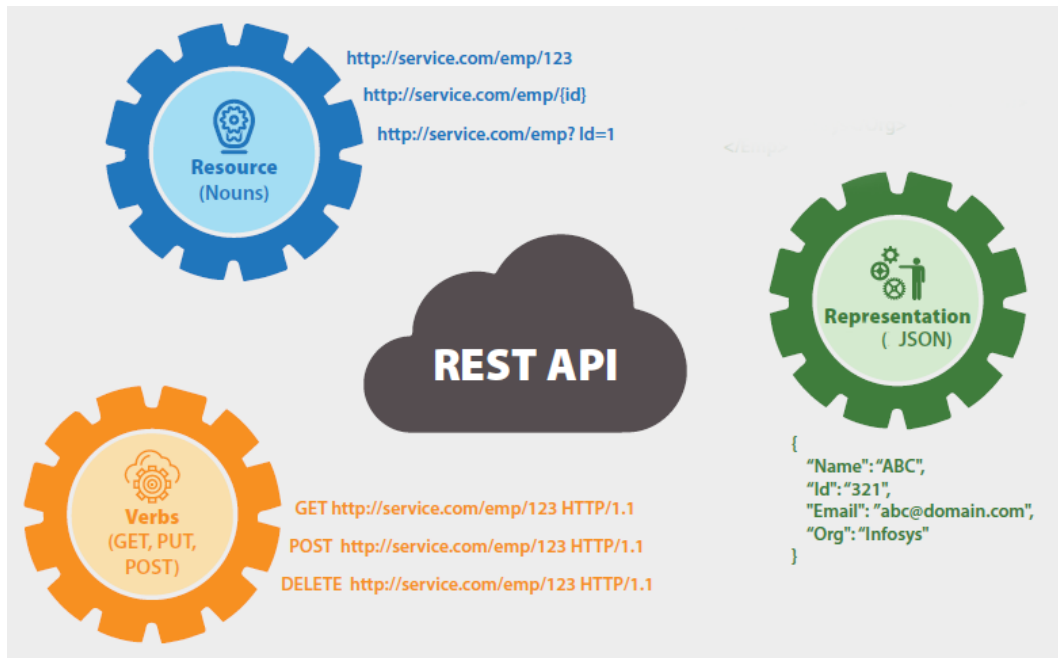
Teknik pengumpulan data dengan cara membuat sebuah forum diskusi, yang berisikan para pengembang aplikasi. Peneliti melakukan diskusi dengan sesama tim pengembang aplikasi Tukutu, guna untuk mengetahui fitur yang dibutuhkan dalam sistem.

3.3 Teknik Analisis Data

Data yang telah diperoleh kemudian dikumpulkan dan dianalisa. Hasil analisa data yang didapatkan akan dilakukan rumusan masalah dengan pengembangan fitur yang cocok sebagai solusi dari masalah tersebut. Solusi yang telah ditemukan diubah menjadi sistem yang terdiri dari fitur – fitur yang telah ditentukan, kemudian dari hasil tersebut dibuat alur sistem pembelian pada aplikasi Tukutu.

3.4 Metode

Dalam pembuatan web service pada sistem pembelian Tukutu, peneliti menggunakan metode REST/RESTFul API. Metode ini cocok digunakan dalam pengembangan web yang menggunakan HTTP verb yang biasa digunakan untuk membuat CRUD (Create Read Update Delete), karena fungsi yang serupa pada HTTP verb yaitu POST, PUT, GET, dan DELETE. Proses yang diterapkan pada pembuatan web service yaitu:



Gambar 3.1: REST Web Service

a) Resource

Resources adalah elemen dasar dari *web*. Ketika bekerja didalam REST, hal pertama yang diidentifikasi adalah *resources* dan menemukan bagaimana *resource - resource* dihubungkan dengan yang lain. Setiap resource memiliki identifikasi yang unik di *web platform* yang disebut *Universal Resource Identifier* (URI) atau pada *web* berupa *Uniform Resource Locator* (URL). Pada sistem tukutu URI yang digunakan menggunakan route milik Lumen yang diberi *prefix* untuk *versioning* API yang digunakan. Tiap resource yang akan diakses memerlukan autentikasi JWT berupa token yang perlu disertakan pada *Request Header* untuk bisa masuk kedalam sistem.

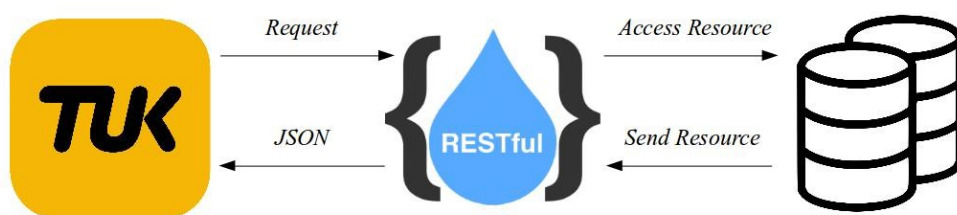
b) Verbs

Verb adalah sebuah aksi HTTP seperti POST, GET, PUT, DELETE. Berikut keterangan mengenai method-method yang digunakan dalam RESTful web service [16]:

- a) GET digunakan untuk mengambil data *resource* dari *server*.
- b) POST digunakan untuk mengirim data *resource* ke *server* atau membuat *resource* yang baru.
- c) PUT digunakan untuk memodifikasi *resource* yang ada di *server*.
- d) DELETE digunakan untuk menghapus data *resource* yang ada di *server*.

c) Representation

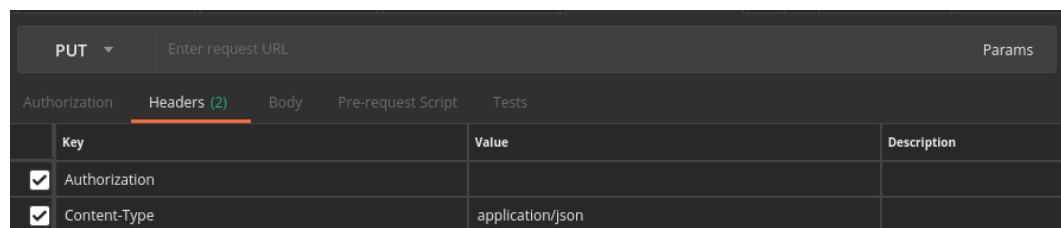
Representasi digunakan untuk menentukan jalan untuk menampilkan *resource – resource* ke *client*. Pada Tukutu memakai *platform* Android yang digunakan oleh pengguna dalam melakukan transaksi yang terhubung dengan *web service*. Pada sistem pembelian Tukutu menggunakan format JSON untuk menerima *resource* dari *web service* pada *platform* Android yang digunakan.



Gambar 3.2: Alur Web Service TUKUTU

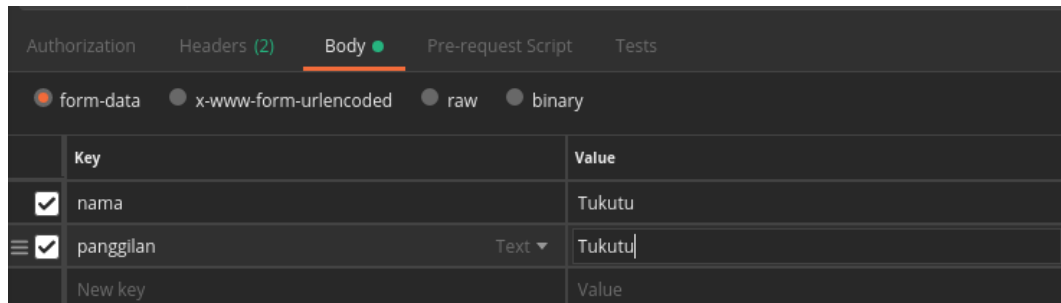
3.5 Eksperimen dan Cara Pengujian Metode

Eksperimen yang dilakukan dengan memanfaatkan *build-in web server* bawaan PHP untuk menjalankan service dan menggunakan POSTMAN sebagai ujicoba REST API yang telah dibuat. Pada saat melakukan pengujian REST API diperlukan *Request Header* tambahan yang berisi **Authorization** dan **Content-Type** yang akan digunakan untuk validasi Request resource pada server. Isi dari **Authorization** berisikan token JWT yang dibuat dari *request* login pada sistem yang kemudian disimpan pada *platform* Android Tukutu yang kemudian akan terus digunakan sebagai hak akses kedalam sistem Tukutu. Sedangkan **Content-Type** digunakan pada *method* PUT HTTP yang digunakan untuk memodifikasi resource, isi dari header tersebut berupa nilai **application/json** sebagai identifikasi data yang dikirim berformat JSON.



Gambar 3.3: Header pada POSTMAN

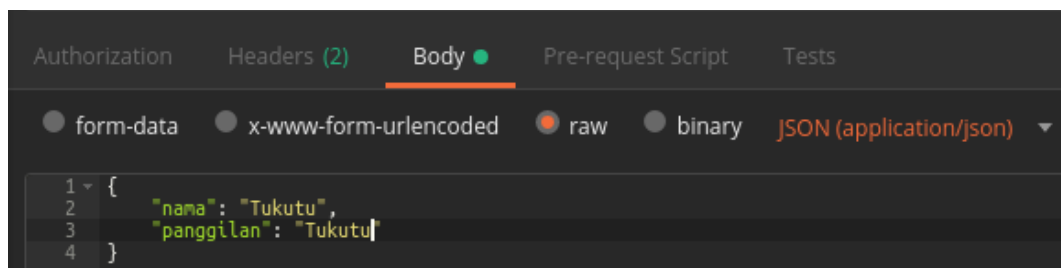
Pada saat *request* method POST atau PUT untuk mengirim resource ke *web service* membutuhkan *Request Body* yang berisi **PARAMS** atau elemen *key* dan *value*. Untuk method PUT resource yang dikirim dalam bentuk format JSON dengan tambahan header **Content-Type**. *Key* dan *value* berisikan nilai yang akan digunakan untuk membuat atau memodifikasi *resource*, *key* yang dimasukan harus sesuai dengan yang telah ditetapkan oleh *web service*.



The screenshot shows the 'Body' tab of a REST client interface. The 'form-data' radio button is selected. Below it, a table lists the request body parameters:

Key	Value
<input checked="" type="checkbox"/> nama	Tukutu
<input checked="" type="checkbox"/> panggilan	Tukutu
New key	Value

Gambar 3.4: Request Body pada method POST



The screenshot shows the 'Body' tab of a REST client interface. The 'raw' radio button is selected, and the content type is set to 'JSON (application/json)'. The request body is a JSON object:

```

1 {
2   "nama": "Tukutu",
3   "panggilan": "Tukutu"
4 }

```

Gambar 3.5: Request Body pada method PUT

Sebagai response dari hasil request menetapkan standar format yang digunakan pada platform Android untuk mengetahui response balik dari web service. Berikut standar format yang digunakan:



```

{
  "status": {
    "success": true,
    "code": 200,
    "message": "OK"
  },
  "result": { }
}

```

Gambar 3.6: Standar format yang digunakan

Penjelasan dari hasil response yang didapat:

- **status**

Berisi status dari hasil request resource yang terdiri dari:

- **success** berisi nilai dengan tipe boolean yang menandakan jika request berhasil atau gagal.
- **code** memiliki nilai dari Response code dari hasil request. Contoh 4xx yang menandakan pada kesalahan request atau 5xx yang menandakan terjadi kesalahan pada server.
- **message** berupa pesan dari hasil request yang berisi pesan kesalahan atau pesan sukses.

- **result**

Merupakan resource dari hasil request yang dapat berupa object atau array yang kemudian ditampilkan.