



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Telaah Literatur

Pada tahap ini dilakukan pembelajaran tentang bagaimana merancang dan membangun *Application Programming Interface*.

2. Perancangan

Pada tahap ini dilakukan perancangan diagram yang bertujuan untuk membantu proses pembuatan API. Pembelajaran terhadap struktur database aplikasi SiCantik juga dilakukan di dalamnya.

3. Pembangunan

Membangun API sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya.

4. Pengujian

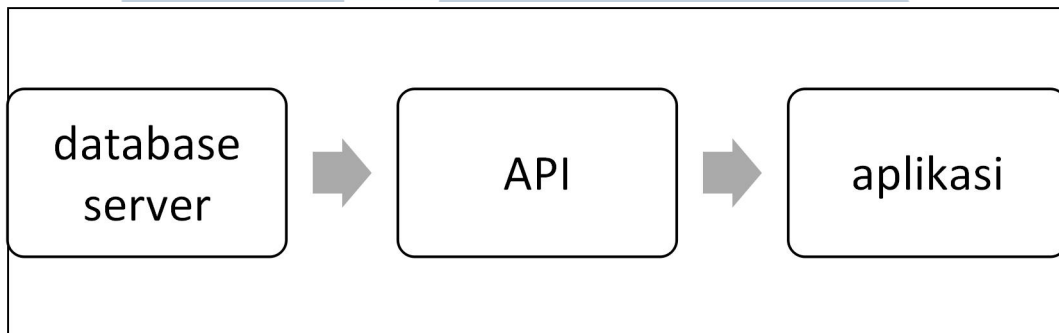
Pada tahap ini dilakukan uji coba API yang telah dibangun untuk mengetahui apabila API telah berjalan dengan benar. Pengujian akan dilakukan menggunakan aplikasi Postman.

5. Evaluasi

Evaluasi akan dilakukan dengan menganalisa hasil pengujian menggunakan Equivalence Partitioning,

### 3.2. Perancangan Sistem

Penelitian diawali dengan membuat perancangan sistem yang akan membantu *user* dalam penyediaan data pasien berupa identitas pasien, yaitu nomor ID, Nama, tanggal lahir, alamat, dan nomor *handphone*.



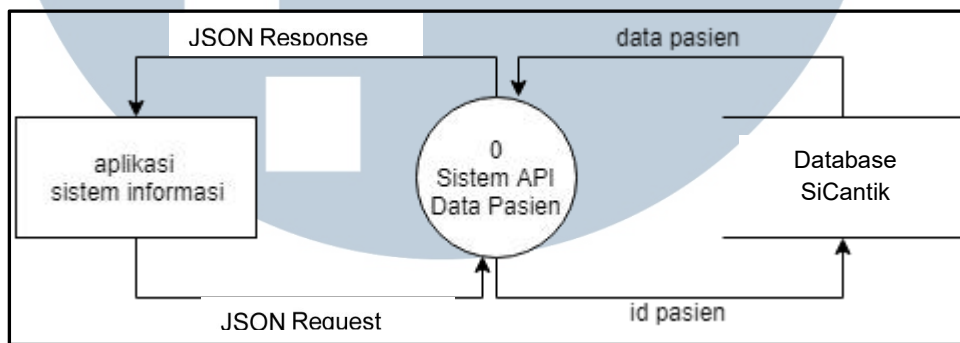
Gambar 3.1 Alur Kerja Sistem

Hal ini dilakukan untuk memastikan sistem bekerja sesuai dengan alur yang telah dirancang, dan memudahkan proses perubahan jika dibutuhkan. Dari gambar 3.1 sapat dijelaskan proses yang dilakukan oleh API untuk mengembalikan data sesuai *request*, yakni:

1. *App (application)* melakukan request data diri, berupa identitas pasien seperti nomor ID, Nama, tanggal lahir, alamat, dan nomor *handphone* sesuai milik pasien.
2. API melakukan query terhadap database pada server.
3. API memproses hasil query.
4. API mengembalikan hasil request pada *app*.

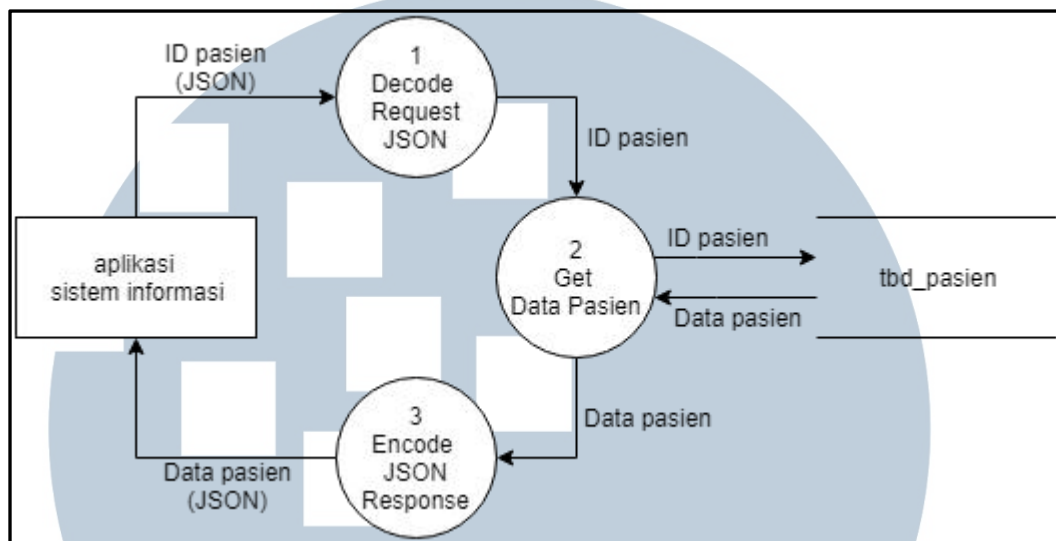
### 3.3 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Adapun DFD dari sistem yang dibuat JSON Response adalah sebagai berikut.



Gambar 3.2 Context Diagram Sistem API

Gambar 3.2 merupakan *Context* Diagram dari sistem rekomendasi yang dirancang. Di dalam diagram tersebut terdapat dua (2) entitas, yaitu aplikasi dan database. Dalam penggunaan sistem API, aplikasi mengirimkan JSON Request yang berisikan data ID Pasien (dalam tabel: cm). Data tersebut akan di proses oleh sistem API dan diteruskan menjadi sebuah *query* untuk menarik data pasien dari *database*. Kemudian data pasien yang merupakan hasil query tersebut akan dikembalikan oleh sistem ke aplikasi dalam bentuk JSON yang bernama JSON Response.



Gambar 3.3 Diagram Level 1 Sistem API

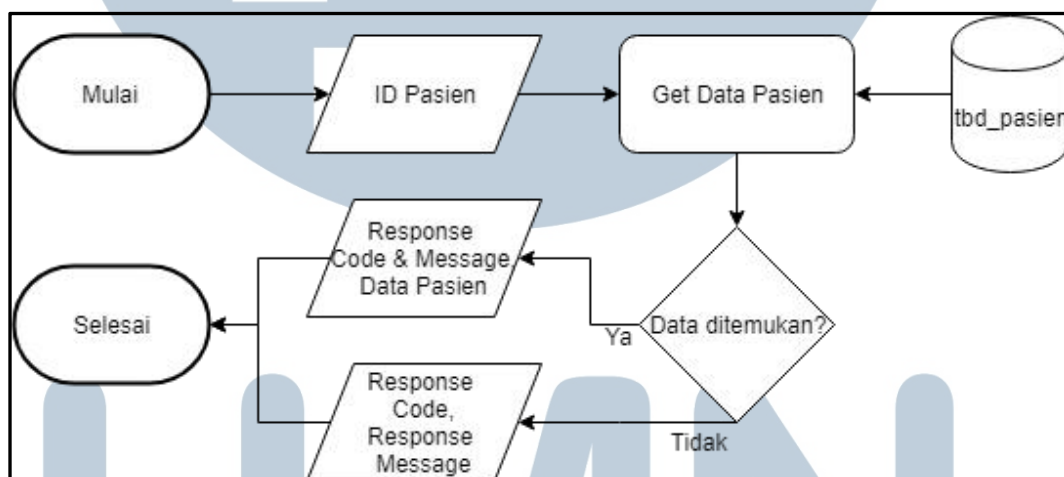
Kemudian pada *Data Flow Diagram* level satu (1) pada Gambar 3.3 terdapat 3 proses utama dalam sistem API. Proses yang pertama yaitu proses *decode* JSON Request yang menerima data ID Pasien yang masih berupa JSON. Data JSON tersebut di proses dengan mekanisme *decode* untuk diperoleh ID Pasien. ID pasien dari hasil *decode* tersebut yang akan digunakan dalam query di dalam proses kedua.

Proses kedua adalah proses Get Data Pasien. Proses ini menerima data ID pasien dari proses *decode*. Data tersebut kemudian digunakan untuk melakukan get pada database, tepatnya pada table *tdb\_pasien*, yang kemudian akan menerima hasil *query* berupa data pasien yang isinya: id, nama, tempat & tanggal lahir, alamat rumah, nomor *handphone*, dan nomor asuransi seorang pasien. Data-data tersebut kemudian diteruskan kepada proses selanjutnya.

Proses selanjutnya, yaitu proses ketiga, proses *encode* yang menerima data-data pasien dari proses get sebelumnya. Dalam proses ini data-data tersebut dirangkum dan diramu menjadi sebuah JSON, untuk dikembalikan kembali ke aplikasi. Data yang dikembalikan tersebut disebut JSON Response.

### 3.4 Flowchart

*Flowchart* digunakan untuk menggambarkan urutan proses yang terjadi secara detail hubungan suatu proses dengan proses yang lain dalam suatu sistem. *Flowchart* digunakan secara umum pada seluruh proses sistem.



Gambar 3.4 Flowchart API

Sistem dimulai dengan menerima input ID pasien dari aplikasi. Seperti yang tertera pada Gambar 3.4, data tersebut kemudian akan digunakan untuk melakukan penarikan data dari database. Jika data ditemukan maka yang akan dikembalikan adalah berupa *response code* dan *response message*, yang kedua menyatakan bahwa data ditemukan, beserta data-data dari pasien itu sendiri. Dan

bila ternyata data tidak ditemukan, maka yang dikembalikan hanyalah *response code* dan *response message* yang menyatakan bahwa data tidak ditemukan.

### 3.4 Struktur Tabel

#### 3.4.1 Tabel Pasien

Tabel 3.1 Tabel tbd\_pasien

No.	Nama Kolom	Tipe Data	Length	Keterangan
1	cm	char	10	Berisi data ID Pasien. Primary key.
2	nama	varchar	150	Nama lengkap pasien
3	tmp_lahir	varchar	20	Tempat lahir pasien
4	tgl_lahir	date		Tanggal lahir pasien
5	alamat	varchar	255	Alamat domisili pasien
6	hp	varchar	20	Nomor handphone pasien
7	no_asuransi	varchar	30	Nomor asuransi pasien

U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A