

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1. Tinjauan Pustaka

Tabel 2.1 Tinjauan Pustaka

Pembeda Sumber	Objek	Metode	Topik	Hasil
Eriza (2013)	TMII	LBS	Wisata	Marker
Widy Sulistianto (2012)	Perguruan Tinggi	Haversine	Pendidikan	Marker
Gia Anggraini (2014)	Wisata Sumsel	JSON	Wisata	Marker, custom map
Nurlaylia (2012)	Stasiun KRL	LBS	Navigasi	Map, marker
Arif Hidayat (2017)	Kuliner Jogja	LBS, Google Direction	Kuliner	Map
Aplikasi yang dikembangkan	Stasiun kereta api	Google Direction, Geofence	Transportasi	Rute, notifikasi

Aplikasi pencarian lokasi Anjungan Taman Mini Indonesia Indah (TMII) berbasis *android* (Eriza Siti Mulyani 2013) menggunakan metode *Location Based Service (LBS)* untuk mengetahui letak dan keterangan objek wisata di TMII yang bias di akses melalui *smartphone Android*. Menggunakan fitur *GPS (Global Positioning System)* dan *A-GPS (Assisted-Global Positioning System)* yang ada pada *smartphone*. Penelitian jurnal ini menggunakan GPS sebagai lokasi user untuk dibandingkan dengan koordinat yang di simpan di database server untuk kemudian ditampilkan menggunakan *google maps*.

Aplikasi pencarian lokasi Perguruan Tinggi di Provinsi Daerah Istimewa Yogyakarta (Widy Sulistianto 2012) ini mengembangkan aplikasi Android untuk

mengetahui posisi user dan jarak user dari perguruan tinggi terdekat dengan kalkulasi rumus *harvesine*. Memanfaatkan fitur GPS untuk menentukan lokasi user yang akan di masukkan ke *database server* dan kemudian dilakukan perhitungan jarak user ke Perguruan Tinggi yang ditampilkan pada *google maps* .

Aplikasi pengenalan pariwisata Sumatera Selatan berbasis android (Gia Anggraini dan Siti Ardianty, 2014). Aplikasi ini menggunakan database *SQLite* dengan sinkronisasi *database MySQL* pada *website* pariwisata Sumatera Selatan. Proses sinkronisasi dilakukan saat aplikasi pengenalan pariwisata Sumatera Selatan terkoneksi dengan internet menggunakan pertukaran data dalam format *JSON (Java script Object Notation)*.

Aplikasi yang dikembangkan yaitu Stasiun Pemberhentian Kereta Prameks Menggunakan Google Maps Berbasis Android. Aplikasi ini menggunakan *GoogleMap* untuk menampilkan peta dan menampilkan rute kereta api menggunakan *GoogleDirection* . Ketika pengguna sampai di stasiun tujuan maka akan muncul alarm yang di *trigger* oleh *Geofence*.

## **2.2. Dasar Teori**

### **2.2.1. Stasiun Kereta Api**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Stasiun adalah “Tempat menunggu bagi calon penumpang kereta api, tempat pemberhentian kereta api” jadi dapat disimpulkan stasiun kereta api adalah tempat pemberhentian kereta api untuk naik turun penumpang kereta api.

### **2.2.2. Prameks**

Prameks adalah singkatan dari Prambanan Ekspres yaitu kereta api berjenis komuter jarak pendek yang melayani koridor perjalanan dari stasisun Solo Balapan kearah barat hingga stasiun Kutoarjo dan termasuk stasiun Yogyakarta. Menurut informasi dari PT.Kereta Api Indonesia (PERSERO) Kereta api Prameks ini sangat populer di wilyah Yogyakarta dan sekitarnya sebagai transportasi umum yang murah dan bebas macet.

### **2.2.3. Location Based Services**

*Location Based Service (LBS)* atau layanan berbasis lokasi adalah sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut. *Location Based Service* dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu

#### 2.2.4. Google Maps Directions API (Route)

*Google Maps Directions API* adalah suatu layanan yang melakukan perhitungan arah diantara beberapa lokasi dengan menggunakan *HTTP request* (Google Developer, 2017)

Parameter yang dibutuhkan oleh *Directions API* adalah nilai garis lintang dan garis bujur dari lokasi asal dan tujuan. Parameter yang bisa ditambahkan antara lain parameter *waypoints* yang digunakan untuk membuat rute perjalanan dari lokasi asal menuju lokasi tujuan. Berikut gambar 2.1 bentuk umum dari sintaks yang digunakan oleh *web service* :

```
https://maps.googleapis.com/maps/api/directions/outputFormat?parameters
```

**Gambar 2.1 Sintaks Umum *Google Direction API***

Sintaks pada gambar 2.1 merupakan URL web service milik google untuk menangani rute dengan menggunakan *polylines*. Pada baris sintaks tersebut terdapat sintaks *outputFormat* yang merupakan format yang akan dikembalikan oleh web service, pada dokumentasi terdapat dua format yang dapat diberikan yaitu dengan format JSON dan XML. Untuk sintaks *parameters* merupakan nilai dari lokasi asal dan tujuan yang dapat berupa nilai letak garis lintang dan bujur, nama kota maupun kode lokasi yang telah ditentukan Google.

#### 2.2.5. GPS

GPS (Global Positioning System) merupakan layanan menggunakan kumpulan satelit yang berjumlah 24 buah yang mengorbit di bumi. GPS akan menemukan posisi pengguna dengan menghitung perbedaan waktu sampainya

sinyal pada suatu satelit menuju satelit lain. Sinyal pada GPS merupakan data dalam bentuk sandi, sehingga perangkat yang bisa menerima data dari GPS adalah perangkat yang memiliki penerima GPS (Manav Singhal and Anupam Shukla, 2012)

### 2.2.6. *JSON Parsing*

Data JSON yang didapatkan dari web service akan diolah dalam aplikasi dengan bentuk Array dan object , sehingga untuk membacanya diperlukan *JSON Parsing*. Data yang suda di-parsing nantinya akan dipakai untuk menampilkan peta dan rute kereta api. Potongan sintaks JSON Parsing pada gambar 2.2.

```
JSONObject overview_polylineJson = jsonRoute.getJSONObject
("overview_polyline");
JSONArray jsonLegs = jsonRoute.getJSONArray("legs");
JSONObject jsonLeg = jsonLegs.getJSONObject(0);
JSONObject jsonDistance = jsonLeg.getJSONObject("distance");
JSONObject jsonDuration = jsonLeg.getJSONObject("duration");
JSONObject jsonEndLocation = jsonLeg.getJSONObject
("end_location");
JSONObject jsonStartLocation =
jsonLeg.getJSONObject("start_location");
```

**Gambar 2.2 Sintaks Parsing JSON**

### 2.2.7. *Geofence*

*Geofence* adalah fitur dalam program perangkat lunak yang menggunakan global positioning system (GPS) untuk menentukan batas - batas geografis. Program yang menggabungkan *geofencing* memungkinkan developer untuk mengatur pemicu sehingga bila perangkat memasuki atau keluar batas-batas (polygon) yang ditetapkan oleh developer maka akan memicu notifikasi pesan

(Google Developer, 2014). Berikut pada gambar 2.3 adalah sintaks umum untuk membuat *geofence*.

```
mGeofenceList.add(new Geofence.Builder()  
    .setRequestId(entry.getKey())  
    .setCircularRegion(  
        entry.getValue().latitude,  
        entry.getValue().longitude,  
        Constants.GEOFENCE_RADIUS_IN_METERS)  
    .setExpirationDuration(Constants.GEOFENCE_EXPIRATION_IN_MILLISECO  
NDS)  
    .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |  
        Geofence.GEOFENCE_TRANSITION_EXIT).build());
```

**Gambar 2.3 Sintaks umum membuat Geofence**