

BAB IV IMPLEMENTASI

4.1 Spesifikasi Sistem

Dengan menggunakan metode LBS (*Location Based Service*) yang dapat digunakan untuk mengetahui posisi berdasarkan titik geografis dapat dibuat program aplikasi GIS (*Geographical Informaton System*) yang bertujuan untuk mencari informasi kuliner yang berguna bagi para pengguna. Berdasarkan POI (*Point of Interest*) yang berupa titik geografis dapat ditentukan lokasi kuliner pada peta.

Dengan penambahan teknologi GPS dan metode A*, aplikasi ini dapat mencari posisi pengguna sehingga dapat dihitung jarak antara pengguna dengan kuliner terdekat yang menjadi tujuan. Di dalam perancangan sistem aplikasi ini terlebih dahulu perlu diketahui analisa – analisa dalam pembuatannya baik dari segi perangkat keras maupun perangkat lunaknya agar mempermudah dalam pembuatan aplikasinya.

4.1.1 Analisa Kebutuhan Perangkat Keras

Perangkat keras merupakan komponen–komponen elektronik yang saling berinteraksi sehingga dapat digunakan untuk memasukkan data, mengolah dan menghasilkan keluaran sesuai dengan data yang diproses. Spesifikasi minimal yang digunakan didalam perancangan aplikasi ini yaitu :

1. Komputer
 - a. Processor Intel Dual Core T2390 1.86 Ghz.
 - b. *Memory* 1 G RAM.
 - c. VGA onboard.
 - d. Hardisk 160 G.
2. *Mobile Device*
 - a. *Operating System* Android 2.2 (Froyo).
 - b. *Processor* ARMv6 600MHz.
 - c. *Memory* 279 MB RAM.
 - d. Layar QVGA (240 X 320) *pixel resolution*.
 - e. *General* HSDPA 7.2Mbps, 900/2100 Mhz, EDGE/GPRS 850/900/1800/1900 Mhz.

4.1.2 Analisa Kebutuhan Perangkat Lunak

Dalam perancangan aplikasi ini digunakan beberapa aplikasi yang dimanfaatkan untuk merancang alur proses sistem. Perancangan alur proses sistem digambarkan dalam bentuk diagram alir, diagram data serta diagram relasi antar data, sebagai contohnya adalah EdrawMax.

Untuk pembangunan aplikasi ini maka dibutuhkan beberapa aplikasi untuk mengoptimalkan kinerjanya. Berikut ini adalah aplikasi yang digunakan sebagai sistem pendukung aplikasi yang dibuat :

1. Sistem operasi, sistem operasi Windows XP sp 3.
2. Java Development Kit 1.6.0_10, *tools* pengembang bahasa pemrograman Java.
3. Eclipse Galileo, pengembangan program dengan lingkungan multi bahasa pemrograman.
4. *Android Development Tools* (ADT 8.0.1 *plug-in*), *plug-in* yang digunakan untuk mengintegrasikan eclipse menjadi lingkungan pengembangan android.
5. *Android SDK 2.2 (Software Development Kit)*, *tools* pengembang program Android.
6. *Map API (Application programming Interface)*, antarmuka program berupa peta digital.
7. *SQLite Manager*, perangkat lunak yang digunakan untuk manajemen basis data.

4.2 Implementasi *GPS Handler*

Implementasi *GPS Handler*, diletakkan pada *class* *Kuliner.java* dan didalam kelas ini akan memanggil sebuah *method* *initMyLocation*. *Method* inilah yang akan menangani posisi pengguna.

```
private void initMyLocation(){  
  
    locationManager = (LocationManager)  
  
    getSystemService(Context.LOCATION_SERVICE);  
    locationManager = new LocationListener() {  
        public void onLocationChanged(Location  
newLocation) {
```

```
tampilkanPosisikeMap(newLocation);

Xawal=newLocation.getLatitude();
Yawal=newLocation.getLongitude();
}

public void onProviderDisabled(String arg0) {
    Toast.makeText(Test1.this, "Provider dinonaktifkan
oleh
                                user, GPS
off",Toast.LENGTH_LONG).show();
}

public void onProviderEnabled(String arg0) {
    Toast.makeText(Test1.this, "Provider diaktifkan
oleh
                                user, GPS
on",Toast.LENGTH_LONG).show();
}

public void onStatusChanged(String arg0, int
arg1,
                                Bundle arg2) {
    Toast.makeText(Test1.this, "Status provider
berubah",Toast.LENGTH_LONG).show();
}
};

locManager.requestLocationUpdates(LocationManager.GPS_P
ROVID
                                ER,0,5, locListener);
}
```

Source Code 4.1 Proses memanggil sebuah *method* *initMyLocation*

Pada *method* `init MyLocation()` akan mendapatkan koordinat GPS dengan bantuan A-GPS atau menggunakan GPS biasa. Pemanggilan koordinat terletak di `getSystemService (Context.LOCATION_SERVICE)` yang selanjutnya akan diproses oleh *method – method* yang berada pada *LocationListener*. *Update* lokasi akan berubah pada setiap waktu 0 detik dan jarak 5 meter.

4.3 Implementasi *KeyApi*

Agar *hardware* dapat terintegrasi dengan google, maka pada tahap ini perlu dilakukan pendaftaran *key* Google Maps Api.

```
<com.google.android.maps.MapView
    android:id="@+id/mapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="true"
    android:apiKey="0YCWvKmvJht5MndJnWcmJvyD20
EEjW70c10mG7Q"
/>
```

Source Code 4.2 proses pendaftaran *key* Google Maps Api

4.4 Implementasi *Map*

Setelah mendapatkan nilai *keyApi* dari google *maps*, maka pengaksesan *map* bisa dilakukan. *Maps* bisa di akses pada *method* `initMapView`.

```
private void initMapView(){
    map=(MapView)findViewById(R.id.mapView);
```

```
controller=map.getController();
map.setSatellite(false);
map.setBuiltInZoomControls(true);
map.getController().setZoom(17);
}
```

Source Code 4.3 Proses pengaksesan *map*

Pengontrolan *map* smuanya dilakukan disini, dari pengesetan tampilan *map* dan juga pengaturan *zoom*.

4.5 Implementasi Internet

Dikarenakan GPS tidak dapat aktif tanpa terhubung dengan internet, maka disini perlu dilakukan penambahan *permission* untuk mengakses internet. Disini juga ada penambahan *library* untuk penanganan google maps. Implementasi ini ditambahkan di AndroidManifest.xml

```
<uses-library
android:name="com.google.android.maps" />

<uses-permission
android:name="android.permission.INTERNET">
</uses-permission>

<uses-permission android:name =
"android.permission.ACCESS_FINE_LOCATION">
</uses-permission>

<uses-permission android:name =
```

```
"android.permission.ACCESS_COARSE_LOCATION">  
</uses-permission>
```

Source Code 4.4 Proses penambahan *permission* untuk mengakses internet

Penjelasan pada script di atas, bahwa MapView membutuhkan koneksi internet untuk bisa mengakses peta dari Google Maps. *COARSE_LOCATION* berfungsi untuk mendeteksi lokasi berdasarkan *GSM provider*. *FINE_LOCATION* berfungsi untuk mendeteksi lokasi berdasarkan GPS satelit.

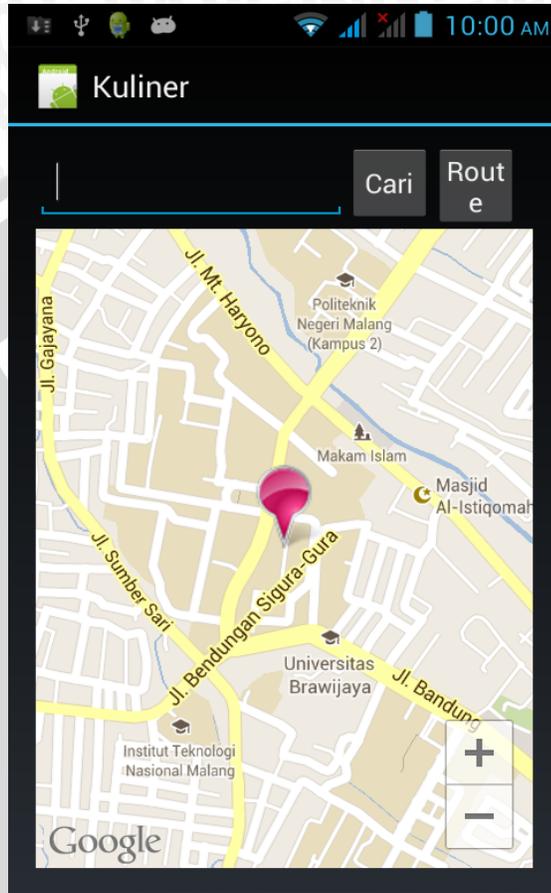
4.6 Implementasi Metode A*

Pada implementasi metode A* akan diletakkan dalam sebuah class bernama *Kuliner.java*. *Class* ini akan menangani perhitungan proses metode A*. Dimana didalam kelas ini terdapat *method* atau prosedur untuk melakukan perhitungan A*. *Method* tersebut nantinya akan dipanggil pada saat *event onClick*. Script lengkapnya terdapat pada lampiran. Setelah melakukan pemanggilan maka akan menghasilkan serangkaian rute dan informasi kriteria kuliner yang dituju.

4.7 Implementasi Sistem

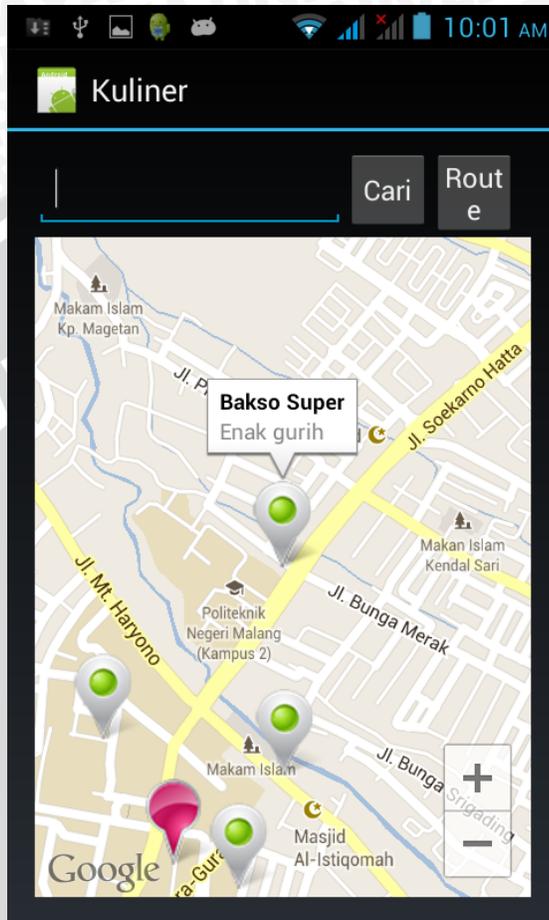
Dalam penerapan aplikasi pencarian jarak terdekat posisi kuliner di kota Malang ini terdapat beberapa tampilan antar muka untuk pengguna di antaranya adalah :

1. Form Tampilan Awal



Gambar 4.1 Form Tampilan Awal

2. *Form List Pencarian*

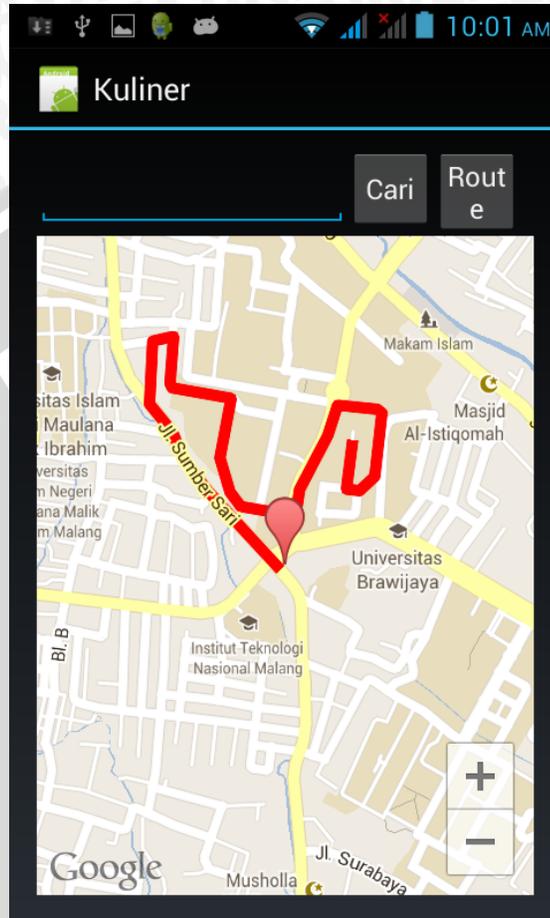


Gambar 4.2 *Form List Pencarian*

Keterangan :

Pada teks area, *user* dapat memasukkan kriteria kuliner yang diinginkan dengan mengetikkan huruf awal dari nama kuliner tersebut. Maka secara otomatis akan muncul nama - nama kuliner yang telah terdaftar.

3. Form Tampil Tujuan



Gambar 4.3 Form Tampil Tujuan

Disini dicontohkan *user* memilih Kuliner Bakso Super dengan menekan tombol route (pencarian), maka terbentuklah penunjuk rute pencarian tersebut.