

## BAB IV

### IMPLEMENTASI DAN EVALUASI

#### 4.1. Implementasi Sistem

Implementasi adalah realisasi dari sebuah aplikasi, pelaksanaan dari suatu rencana, ide, model, algoritma atau kebijakan. Didalam ilmu komputer, implementasi merupakan realisasi dari spesifikasi teknis atau algoritma sebagai sebuah program, komponen perangkat lunak atau sistem komputer lainnya melalui pemrograman dan pengembangan. Berikut adalah penjelasan dari proses-proses yang ada pada aplikasi berdasarkan rancangan yang dibuat pada bab tiga.

##### 4.1.1 Implementasi Penerapan Fungsi GPS

Fungsi GPS selalu dibutuhkan pada aplikasi ini yaitu untuk menentukan keberadaan posisi user. pada proses penentuan posisi user, aplikasi akan mengambil data koordinat yang terdapat pada perangkat GPS. Dalam pengambilan data koordinat dari GPS diperlukan *class* yang mengimplementasikan *location listener* dan sebuah atribut *location manager*.

*Location listener* berfungsi sebagai penangkap data koordinat yang diterima GPS dari satelit. Pada *class location listener* dapat ditentukan apa yang harus dilakukan aplikasi ketika terjadi perpindahan lokasi, terjadi perubahan status GPS dan ketika GPS diaktifkan atau di non-aktifkan. Sedangkan atribut *location manager* memiliki fungsi sebagai pengatur waktu aplikasi untuk mengambil data koordinat dari GPS. Pada contoh dibawah ini telah diatur aplikasi akan meng-

*update* data koordinat setiap 1 detik atau terjadi perpindahan dengan jarak 10 meter dari koordinat sebelumnya.

```
locListener = new myLocationListener();
locManager=(LocationManager) getSystemService(Context.LOCATION_SERVICE);
locManager.requestLocationUpdates (LocationManager.GPS_PROVIDER
    , 1000, 10, locListener);

class myLocationListener implements LocationListener{
    public void onLocationChanged(Location arg0) {
        // TODO Auto-generated method stub
    }
    public void onProviderDisabled(String arg0) {
        // TODO Auto-generated method stub
    }
    public void onProviderEnabled(String arg0) {
        // TODO Auto-generated method stub
    }
    public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
        // TODO Auto-generated method stub
    }
}
```

Pada aplikasi yang dibuat dalam menentukan nama jalan juga menggunakan fungsi GPS. Penentuan jalan menggunakan metode titik terdekat. Jadi pada suatu jalan diberi titik koordinat sejumlah 10 titik. Dengan menggunakan GPS jika jarak dari koordinat GPS <50 meter terhadap salah satu titik dari 10 titik pada suatu jalan maka koordinat sesuai GPS berada di jalan tersebut.

#### 4.1.2 Implementasi Penerapan Proses Penggambaran Simbol

Dalam proses penggambaran simbol, aplikasi akan menggunakan *class item overlay*. Pada *class* ini terdapat beberapa *method* seperti contoh dibawah ini.

```

public AddItemizedOverlay(Drawable defaultMarker) {
    super(boundCenterBottom(defaultMarker));
}

public AddItemizedOverlay(Drawable defaultMarker, Context context) {
    this(defaultMarker);
    this.context = context;
}

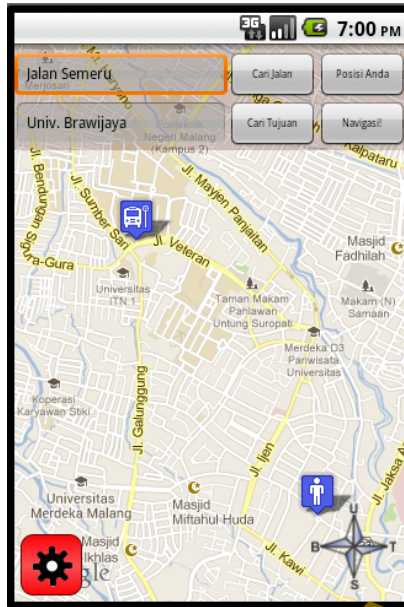
@Override
protected OverlayItem createItem(int i) {
    return mapOverlays.get(i);
}

@Override
public int size() {
    return mapOverlays.size();
}

public void addOverlay(OverlayItem overlay) {
    mapOverlays.add(overlay);
    this.populate();
}

```

Pada contoh *class* di atas terdapat dua buah *constructor* yang digunakan untuk pemanggil *class* ini. *Method create item* digunakan untuk membuat atau menyiapkan item yang akan digambar. Item tersebut berupa file berbentuk gambar. *Method size* digunakan untuk menghitung jumlah gambar yang telah digambar pada peta. Sedangkan *method addoverlay* digunakan untuk menambahkan atau menggambar simbol pada peta. Pada Gambar 4.1 dapat dilihat contoh penggambaran simbol pada peta.



Gambar 4.1 Contoh Penggambaran Simbol

#### 4.1.3 Implementasi Penerapan Proses Penggambaran Rute

Dalam proses penggambaran rute, aplikasi menggunakan *class route overlay*. Proses penggambaran rute ini memiliki cara yang hampir sama dengan proses penggambaran simbol. Hanya saja pada class ini menambahkan method yang digunakan untuk menggambar garis. Berikut adalah beberapa method yang ditambahkan pada *class route overlay*.

```

public RouteOverlay(final Route route, final int defaultColour) {
    super();
    routePoints = route.getPoints();
    colour = defaultColour;
    path = new Path();
    p = new Point();
    paint = new Paint();
}

@Override
public final void draw(final Canvas c, final MapView mv,
    final boolean shadow) {
    super.draw(c, mv, shadow);
    paint.setColor(colour);
    paint.setAlpha(ALPHA);
    paint.setAntiAlias(true);
    paint.setStrokeWidth(STROKE);
    paint.setStyle(Paint.Style.STROKE);
    redrawPath(mv);
    c.drawPath(path, paint);
}

private void redrawPath(final MapView mv) {
    final Projection prj = mv.getProjection();
    path.rewind();
    final Iterator<GeoPoint> it = routePoints.iterator();
    prj.toPixels(it.next(), p);
    path.moveTo(p.x, p.y);
    while (it.hasNext()) {
        prj.toPixels(it.next(), p);
        path.lineTo(p.x, p.y);
    }
    path.setLastPoint(p.x, p.y);
}
}

```

Pada contoh class di atas *method draw* berfungsi sebagai pembuat garis sedangkan *method redrawPath* berfungsi sebagai penggambar garis pada peta. Rute pada map merupakan sebuah garis yang digambar dari titik awal ke titik tujuan. Dalam penggambaran garis ini membutuhkan sebuah perintah khusus yang telah disediakan google map.

<http://maps.googleapis.com/maps/api/directions/json?origin=koordinat awal&destination=koordinat tujuan,&sensor=true>

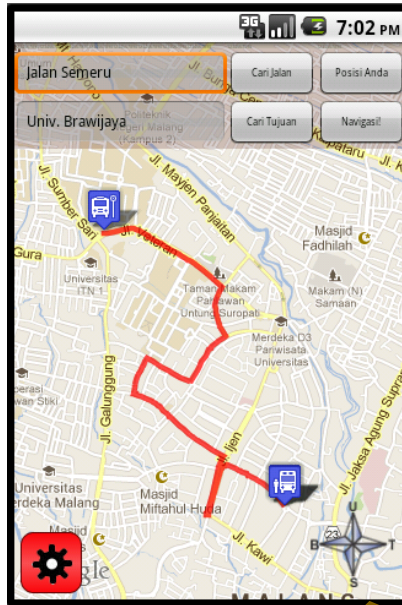
#### 4.1.4 Implementasi Penerapan Proses Navigasi

Dalam proses navigasi, aplikasi menggunakan fungsi dari *class item overlay* dan *class route overlay* secara bersamaan. Titik koordinat pada proses navigasi ini menggunakan data koordinat dari database. Pada penentuan titik koordinat yang

akan digunakan, aplikasi akan melakukan *query* ke dalam database. Aplikasi akan melakukan *query* setelah user menentukan titik koordinat awal dan titik koordinat tujuan.

Setelah mendapatkan titik koordinat awal dan tujuan, akan didapat nama jalan berdasarkan titik koordinat tersebut. Setelah didapat nama jalan awal dan tujuan, aplikasi melakukan *query* untuk mencari mikrolet apa yang melewati jalan tujuan. setelah didapat beberapa mikrolet, aplikasi akan mnyeeksi mikrolet yang didapat apakah mikrolet tersebut juga melewati jalan awal. Jika terdapat jalan awal dan jalan tujuan pada satu buah mikrolet maka aplikasi akan menggunakan titik koordinat mikrolet tersebut untuk menggambar rute dari titik koordinat awal ke titik koordinat tujuan.

Sedangkan jika tidak terdapat mikrolet yang melewati jalan awal dan jalan tujuan, maka aplikasi akan melakukan *query* untuk mencari mikrolet mana yang melewati jalan awal. Setelah didapat data tersebut, aplikasi akan melakukan *query* untuk menghubungkan jalan yang sama-sama dilewati mikrolet awal dan mikrolet tujuan. Setelah didapat mikrolet yang menghubungkan titik koordinat awal dan titik koordinat tujuan, aplikasi akan menggambar rute mikrolet teersbut dengan perbedaan warna garis rute. Pada Gambar 4.2 dapat dilihat contoh penggambaran rute navigasi.

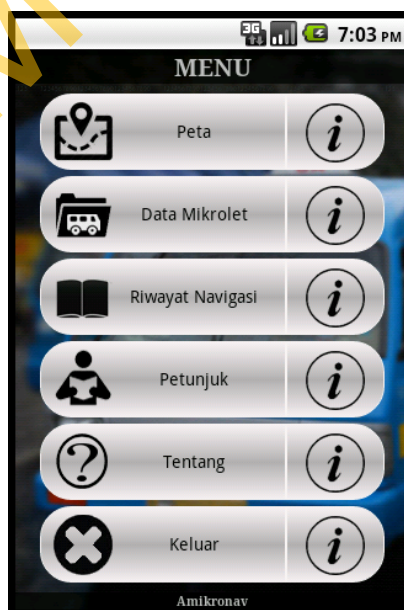


Gambar 4.2 Contoh Penggambaran Rute Navigasi

#### 4.1.5 Implementasi Antarmuka

Berikut adalah beberapa antarmuka yang telah diimplementasikan sesuai dengan desain antarmuka pada subbab 3.3.

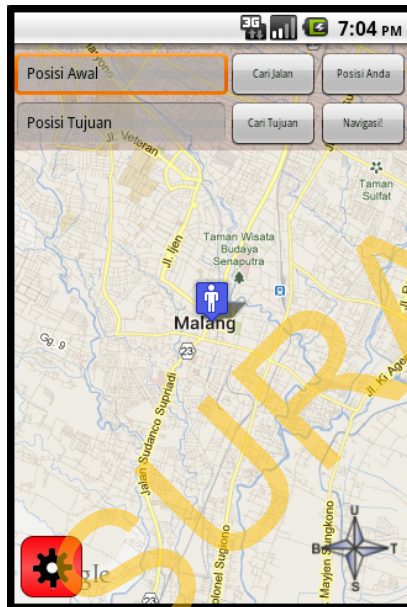
##### a. Tampilan Form Menu Utama



Gambar 4.3 Tampilan Menu Utama

Gambar 4.3 merupakan tampilan menu utama aplikasi. Pada menu utama tersebut terdapat enam pilihan menu dimana pada sebelah kanan tombol menu terdapat info mengenai tombol menu tersebut. Apabila tombol info di klik maka akan menampilkan fungsi dari tombol menu.

b. Tampilan Form Peta



Gambar 4.4 Tampilan Peta

Gambar 4.4 merupakan gambar tampilan peta yang ditampilkan aplikasi setelah user memilih menu “Peta” pada menu utama. Pada tampilan ini memiliki berbagai macam fungsi yang dapat digunakan oleh user, yaitu fungsi penentuan posisi user, fungsi cari jalan, fungsi cari tujuan dan fungsi navigasi. Setiap user mengganti posisi awal dan posisi tujuan, pada *textbox* posisi jalan dan posisi tujuan akan berganti dengan nama jalan posisi tersebut.



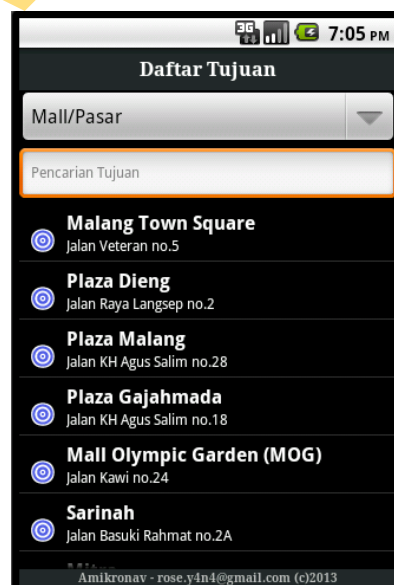
c. Tampilan Form Daftar Jalan



Gambar 4.5 Tampilan Daftar Jalan

Gambar 4.5 merupakan tampilan daftar jalan yang ditampilkan aplikasi setelah user menekan tombol “cari jalan” pada tampilan peta. Pada tampilan ini user dapat memilih salah satu jalan untuk dijadikan sebagai posisi awal user ketika user akan melakukan navigasi.

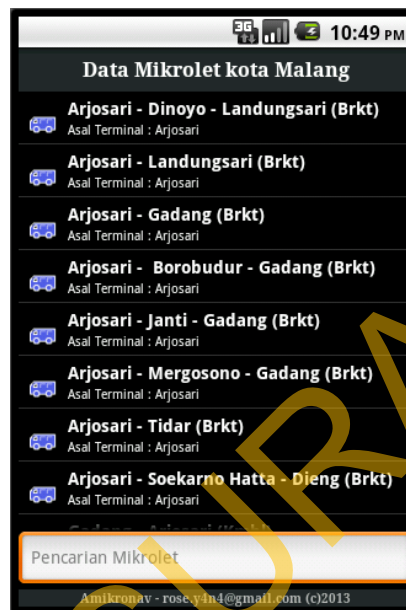
d. Tampilan Form Daftar Tujuan



Gambar 4.6 Tampilan Daftar Tujuan

Gambar 4.6 merupakan tampilan daftar tujuan yang ditampilkan aplikasi setelah user menekan tombol “cari tujuan” pada tampilan peta. User dapat memilih salah satu tujuan untuk dijadikan sebagai posisi tujuan untuk proses navigasi.

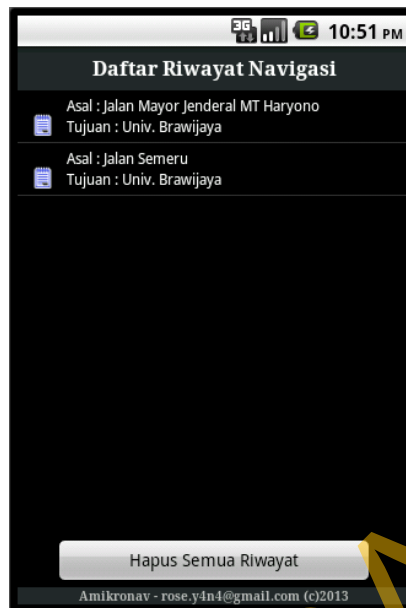
e. Tampilan Form Daftar Mikrolet



Gambar 4.7 Tampilan Daftar Mikrolet

Gambar 4.7 merupakan tampilan daftar mikrolet yang ditampilkan oleh aplikasi setelah user memilih menu “Data Mikrolet” pada menu utama. User dapat memilih salah satu mikrolet untuk ditampilkan informasi mengenai jalan yang dilewati oleh mikrolet. Selain itu dari info tersebut user juga dapat menampilkan rute mikrolet pada peta.

f. Tampilan Riwayat Navigasi



Gambar 4.8 Tampilan Riwayat Navigasi

Gambar 4.8 merupakan tampilan riwayat navigasi yang pernah dilakukan oleh user. Tampilan ini dapat dilihat setelah user menekan tombol “Riwayat Navigasi” pada menu utama aplikasi.

## 4.2. Pengujian Aplikasi

Setelah melakukan proses implementasi *class* dan implementasi antarmuka, maka proses selanjutnya adalah proses pengujian aplikasi dengan tujuan untuk mengetahui apakah aplikasi yang sudah dibuat dapat menghasilkan *output* sesuai dengan yang diharapkan.

### 4.2.1 Tujuan Pengujian

Terdapat beberapa tujuan pengujian sebagai berikut :

- a. Menguji penggunaan fungsi GPS untuk menentukan posisi awal.
- b. Menguji penggunaan fungsi penggambaran simbol dan rute.

- c. Menguji penggunaan fungsi navigasi untuk melakukan navigasi.

#### 4.2.2 Data Uji

Berikut ini adalah data uji yang digunakan dalam pengujian:

- a. Memilih “Peta” pada menu utama aplikasi agar aplikasi menampilkan peta kota Malang. Kemudian menekan tombol “Posisi User”.
- b. Memilih “Data Mikrolet” pada menu utama kemudian memilih salah satu mikrolet untuk ditampilkan rutenya.
- c. Memilih “Peta” pada menu utama kemudian menekan tombol “Cari Jalan” untuk mencari jalan. Kemudian menekan tombol “Cari Tujuan” untuk mencari tujuan. Setelah itu, menekan tombol “Navigasi” untuk melakukan navigasi manual.
- d. Memilih “Peta” pada menu utama kemudian menekan tombol “Posisi User” untuk menentukan posisi berdasarkan GPS. Kemudian menekan tombol “Cari Tujuan” untuk mencari tujuan. setelah itu, menekan tombol “Navigasi” untuk melakukan navigasi dengan GPS.

#### 4.2.3 Kasus Uji

Berdasarkan tujuan pengujian yang didefinisikan pada subbab 4.2.1, maka terdapat 4 buah kasus untuk pengujian dengan data pengujian yang telah didefinisikan pada subbab 4.2.2.

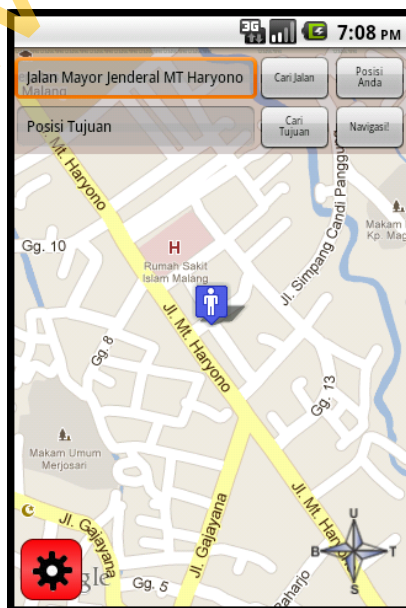
- a. Kasus uji 1, menguji fungsi GPS untuk menentukan posisi user yaitu pengujian yang dilakukan untuk mengetahui *class location listener* dan *location manager* mampu mendapatkan titik koordinat user dari GPS.

- b. Kasus uji 2, menguji fungsi *class item overlay* dan *route overlay* yaitu pengujian yang dilakukan untuk mengetahui apakah class tersebut dapat menggambar pada peta dengan benar sesuai titik koordinat.
- c. Kasus uji 3, menguji fungsi navigasi dengan memilih titik awal secara acak yaitu pengujian yang dilakukan untuk mengetahui fungsi navigasi dapat dilakukan dengan penggunaan *class item overlay* dan *class route overlay* yang kemudian dapat dilihat hasil navigasinya pada peta.
- d. Kasus uji 4, menguji fungsi navigasi dengan titik awal posisi user yaitu pengujian yang dilakukan untuk mengetahui fungsi navigasi dapat dilakukan dengan penggunaan fungsi GPS, *class item overlay* dan *class route overlay* yang kemudian dapat dilihat hasil navigasinya pada peta.

#### 4.2.4 Hasil Pengujian

Berikut ini adalah hasil pengujian dari kasus uji yang dilakukan.

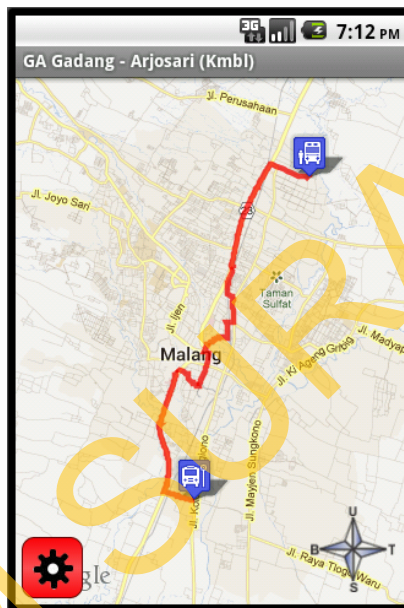
- a. Kasus uji 1



Gambar 4.9 Hasil Pengujian Kasus Uji 1

Pada Gambar 4.9 dapat diketahui bahwa penggunaan *class location listener* dan *location manager* berjalan dengan sukses karena dapat menunjukkan posisi user berdasarkan koordinat yang diambil dari perangkat GPS. Selain itu, pada pengujian ini juga menampilkan bahwa penggunaan *class item overlay* juga sukses karena aplikasi sukses menampilkan posisi user melalui simbol.

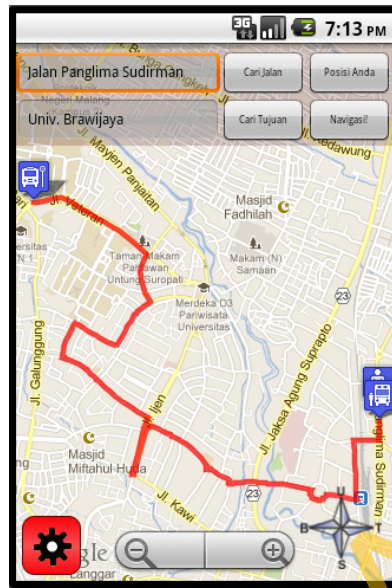
b. Kasus uji 2



Gambar 4.10 Hasil Pengujian Kasus Uji 2

Pada Gambar 4.10 dapat diketahui bahwa *class item overlay* dan *class route overlay* dapat digunakan secara bersamaan. Dalam penggunaan dua class tersebut juga berjalan sukses karena aplikasi dapat menampilkan simbol dan rute mikrolet secara bersamaan.

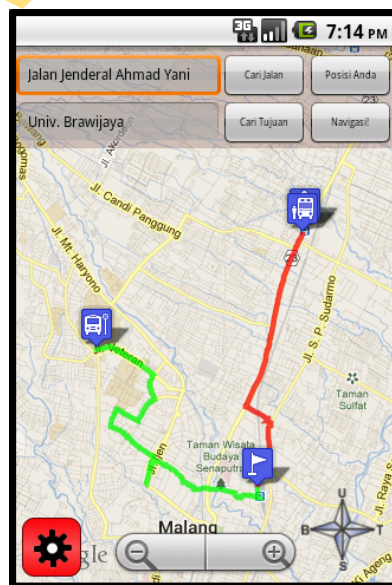
## c. Kasus uji 3



Gambar 4.11 Hasil Pengujian Kasus Uji 3

Pada Gambar 4.11 dapat diketahui bahwa penggunaan fungsi navigasi dapat berjalan sukses. Fungsi navigasi akan menghasilkan beberapa titik koordinat yang akan digunakan oleh *class item overlay* dan *class route overlay* untuk menggambar simbol dan rute pada peta.

## d. Kasus uji 4



Gambar 4.12 Hasil Pengujian Kasus Uji 4

Gambar 4.12 merupakan hasil pengujian kasus uji 4 yaitu tentang penggunaan berbagai fungsi aplikasi secara bersamaan. Pada hasil pengujian ini menunjukkan aplikasi dapat mengkoordinasikan penggunaan fungsi GPS untuk menentukan posisi user kemudian melakukan fungsi navigasi ke titik tujuan dengan penggunaan *class item overlay* dan *class route overlay* sebagai penggambar simbol dan rute pada peta.

#### 4.2.5 Hasil Pengujian pada *Smartphone* Android

Proses uji coba ini dilakukan untuk mengetahui tingkat kompatibilitas aplikasi. Uji coba ini dilakukan pada empat versi *operating system* android. Daftar *operating system* android yang digunakan dapat dilihat pada tabel 4.1. Keempat versi android tersebut merupakan versi android yang banyak digunakan oleh masyarakat.

Tabel 4.1 Daftar Versi OS Android

No.	Versi OS Android
1	Android 2.2 (Froyo)
2	Android 2.3.6 (Gingerbread)
3	Android 4.0 (Ice Cream Sandwich)
4	Android 4.1 (Jelly Bean)

Uji coba dilakukan pada *smartphone* android dengan versi *operating system* android sesuai tabel 4.1. uji coba ini bertujuan untuk mengetahui tingkat kompatibilitas masing-masing fungsi dari aplikasi apakah aplikasi dapat berjalan sempurna di beberapa versi android. Hasil uji coba dapat dilihat pada tabel 4.2.



Tabel 4.2 Hasil Uji Coba Aplikasi pada Smartphone Android

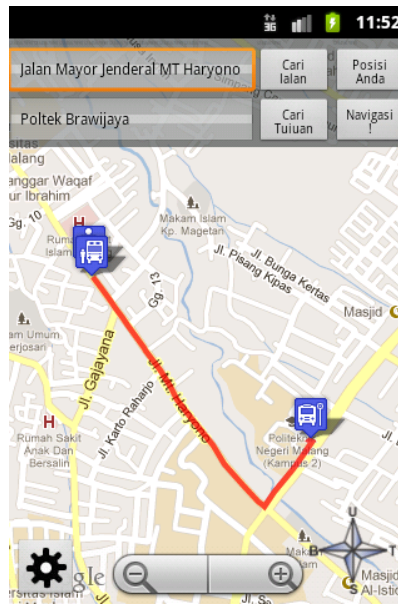
Uji Coba \ Merk	Android 2.2	Android 2.3.6	Android 4.0	Android 4.1
Instalasi Aplikasi	OK	OK	OK	OK
Penentuan Posisi User	OK	OK	OK	OK
Navigasi GPS	OK	OK	OK	OK
Navigasi Manual	OK	OK	OK	OK
Data Mikrolet	OK	OK	OK	OK

Berikut adalah beberapa tampilan hasil pengujian aplikasi menggunakan smartphone android. Tampilan yang dihasilkan oleh smartphone ini diambil dengan menggunakan aplikasi tambahan yang berfungsi untuk meng-capture tampilan aplikasi pada smartphone.



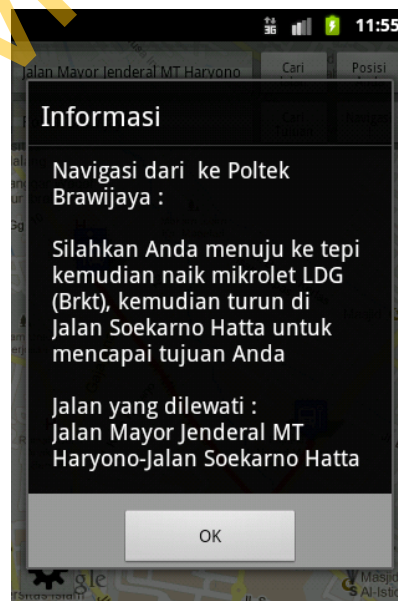
Gambar 4.13 Tampilan Posisi User pada Smartphone Android

Pada Gambar 4.13 merupakan hasil capture ketika menggunakan aplikasi ini dengan smartphone android pada lokasi sebenarnya. Pada gambar tersebut menunjukkan bahwa posisi user berada di jalan Mt. Haryono. Posisi tersebut diambil berdasarkan koordinat GPS setelah menekan tombol “Posisi Anda”.



Gambar 4.14 Tampilan Navigasi pada Smartphone Android

Pada Gambar 4.14 merupakan hasil capture ketika mencoba melakukan navigasi ke tujuan yaitu Poltek Brawijaya. Dari posisi awal di jalan Mt. Haryono user mencoba melakukan navigasi dengan mikrolet ke Poltek Brawijaya. Kemudian aplikasi memberikan rute navigasi seperti gambar di atas.



Gambar 4.15 Tampilan Info Navigasi pada smartphone Android

Pada Gambar 4.15 menunjukkan informasi navigasi ketika melakukan navigasi dari posisi awal jalan Mt. Haryono ke tujuan Poltek Brawijaya. Informasi yang diberikan aplikasi yaitu user harus naik mikrolet LDG pada tepi jalan Mt. Haryono kemudian turun di jalan Soekarno Hatta untuk tiba di Poltek Brawijaya. Ketika pengujian, aplikasi memberikan informasi yang sesuai dengan keadaan sebenarnya.

### **4.3. Evaluasi Sistem**

Berdasarkan uji coba yang dilakukan dapat disimpulkan kemampuan dan kelemahan aplikasi sebagai berikut :

#### **4.3.1 Kemampuan Aplikasi**

Kemampuan dari aplikasi yang dibangun antara lain :

1. Aplikasi dapat menentukan lokasi user berdasarkan koordinat GPS pada *device* Android.
2. Aplikasi dapat memberikan informasi mengenai mikrolet di kota Malang serta memberi informasi mengenai jalan dan rute yang dilewati mikrolet.
3. Aplikasi dapat melakukan navigasi dengan mikrolet dan memberikan informasi navigasi pada user.

#### **4.3.2 Kelemahan Aplikasi**

Kelemahan dari aplikasi yang dibangun antara lain :

1. Karena aplikasi yang dibangun menggunakan peta dari Google Map sehingga dalam penggunaannya selalu membutuhkan koneksi internet. Jika

koneksi internet pada *device* Android milik user mengalami masalah maka aplikasi akan mengalami gangguan dalam memberikan hasil navigasi.

STIKOM SURABAYA