



REPUBLIK INDONESIA  
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201975355, 10 Oktober 2019

## Pencipta

Nama : **Muhammad Izzuddin Mahali, M.Cs., Dr. Ir. Eko Marpanaji, M.T., , dkk**

Alamat : Krapyak RT08 RW20 Triharjo Sleman Yogyakarta, Sleman, Di Yogyakarta, 55514

Kewarganegaraan : Indonesia

## Pemegang Hak Cipta

Nama : **LPPM UNY**

Alamat : Jl Colombo No 1 Caturtunggal Depok Sleman, Sleman, Di Yogyakarta, 55281

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **Aplikasi Android Smart Traffic Light Menggunakan Pemrograman Java**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 14 September 2018, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000158318

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL



Dr. Freddy Harris, S.H., LL.M., ACCS.  
NIP. 196611181994031001

## LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhammad Izzuddin Mahali, M.Cs.	Krapyak RT08 RW20 Triharjo Sleman Yogyakarta
2	Dr. Ir. Eko Marpanaji, M.T.	Polaman RT017 Argorejo Sedayu Bantul
3	Ir. Satriyo Agung Dewanto, M.Pd.	Badran JT I/722 RT047 RW011 Bumijo Jetis Kota Yogyakarta
4	Dr. Umi Rochayati, M.T.	Sedan RT005/RW034 Sariharjo Ngaglik Sleman Yogyakarta
5	Bekti Wulandari, M.Pd.	Pundung Pedukuhan V RT020/RW010 Banaran Galur Kulon Progo
6	Nur Hasanah, M.Cs.	Perum Mranggen A 12 RT002/RW024 Sinduadi Mlati Sleman Yogyakarta





# 2019

Muhammad Izzuddin Mahali, M.Cs.

Dr. Ir. Eko Marpanaji, M.T.

Ir. Satriyo Agung Dewanto, M.Pd.

Dr. Umi Rochayati, M.T.

Bekti Wulandari, M.Pd.

Nur Hasanah, M.Cs.

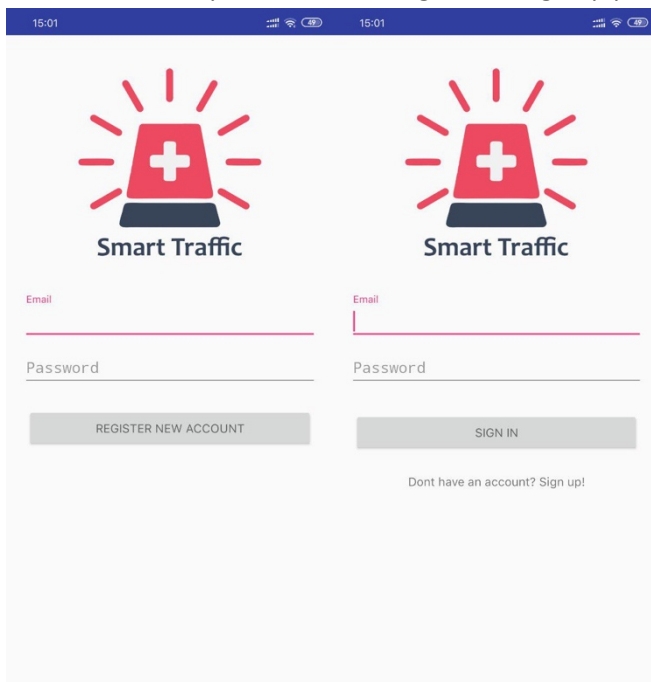
## **Aplikasi Android Smart Traffic Light Menggunakan Pemrograman Java**

## SMART TRAFFIC LIGHT MENGUNAKAN PEMROGRAMAN JAVA

Aplikasi Android Smart Traffic Light merupakan aplikasi berbasis android pendukung sistem besar bernama Smart Traffic Light for High Priority Vehicle. Sistem ini merupakan sistem yang diciptakan dengan tujuan untuk mempersingkat waktu tempuh kendaraan prioritas. Aplikasi yang digunakan oleh driver pengendara kendaraan prioritas akan melakukan request terhadap lampu APILL yang akan dilalui melalui basis komunikasi Internet of Things. Aplikasi akan mengirimkan status jarak ke penyimpanan awan kemudian lampu APILL yang sudah terintegrasi dengan sistem Smart Traffic Light akan membaca status jarak tersebut. Apabila kendaraan prioritas yang akan melintas sudah memasuki Emergency Area maka lampu APILL pada arah datang kendaraan prioritas akan dinyalakan dengan nyala warna hijau sehingga penumpukan antrian kendaraan yang menunggu giliran melintas sudah terurai dan kendaraan prioritas dapat melaju tanpa menunggu lama di simpang. Berikut ini adalah tampilan petunjuk penggunaan dan source code program menggunakan bahasa java.

### 1) Login dan Register

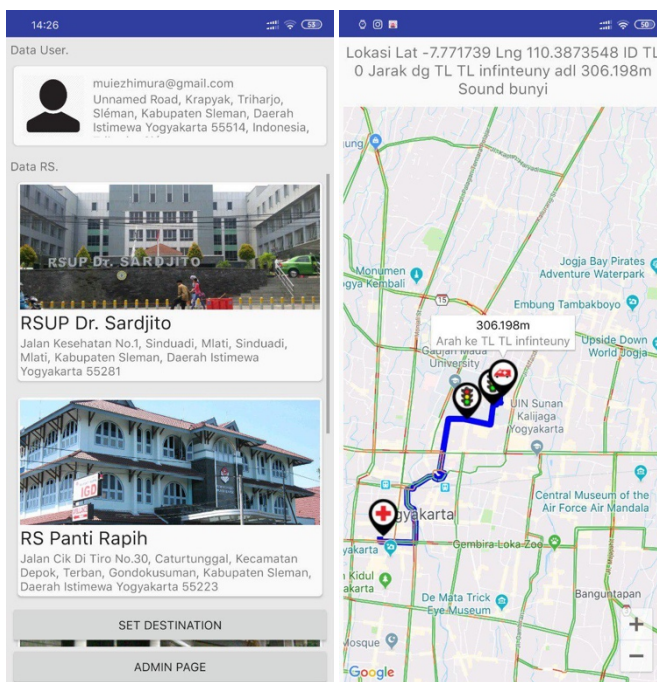
Halaman login akan muncul pertama kali ketika aplikasi dibuka. Apabila pengguna sudah memiliki akun dapat langsung memasukkan akun kemudian klik tombol SIGN IN. Pengguna yang belum memiliki akun dapat melakukan registrasi /Sign up pada menu yang sudah tersedia.



## 2) Landing Page

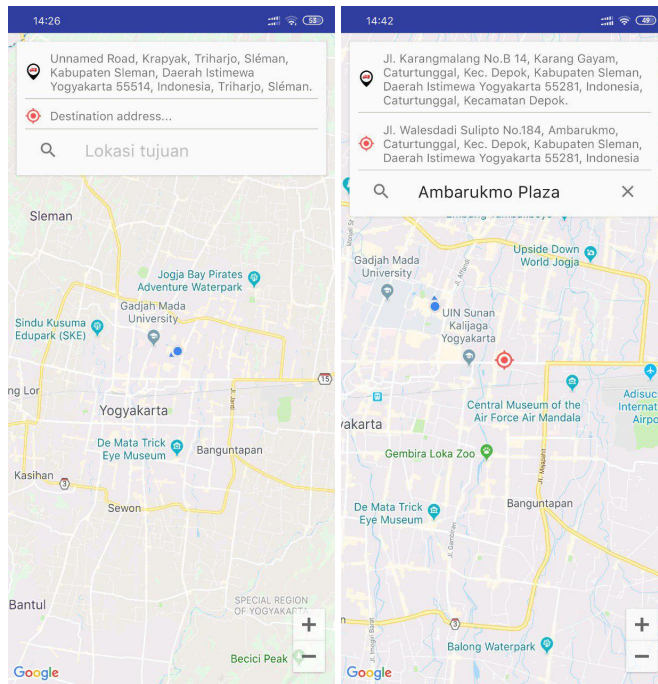
Setelah melakukan login terdapat halaman yang berisikan data profile dan lokasi pengguna dalam ordinat GPS. Di bagian bawah terdapat daftar Rumah Sakit serta alamat lengkap. Pada bagian bawah halaman terdapat halaman untuk untuk menentukan titik ordinat secara manual pada menu SET DESTINATION dan menu seting pada halaman (ADMIN PAGE).

Pengguna dapat melakukan pilihan melalui daftar Rumah Sakit Rujukan. Apabila pengguna sudah memilih daftar Rumah Sakit Rujukan yang sudah tersedia sistem akan mencari Rute tercepat dan ditampilkan dalam peta digital dengan jalur berwarna biru. Aplikasi akan mengidentifikasi letak ordinat Smart Traffic Light APILL yang akan dilalui. Jarak pengguna dengan Smart Traffic Light APILL terdekat akan dimunculkan dalam dashboard berikut seterusnya. Apabila pengguna sudah memasuki Area Emergency, Aplikasi akan mengirimkan data arah kedatangan pengguna dan jarak terhadap Smart Traffic Light APILL.



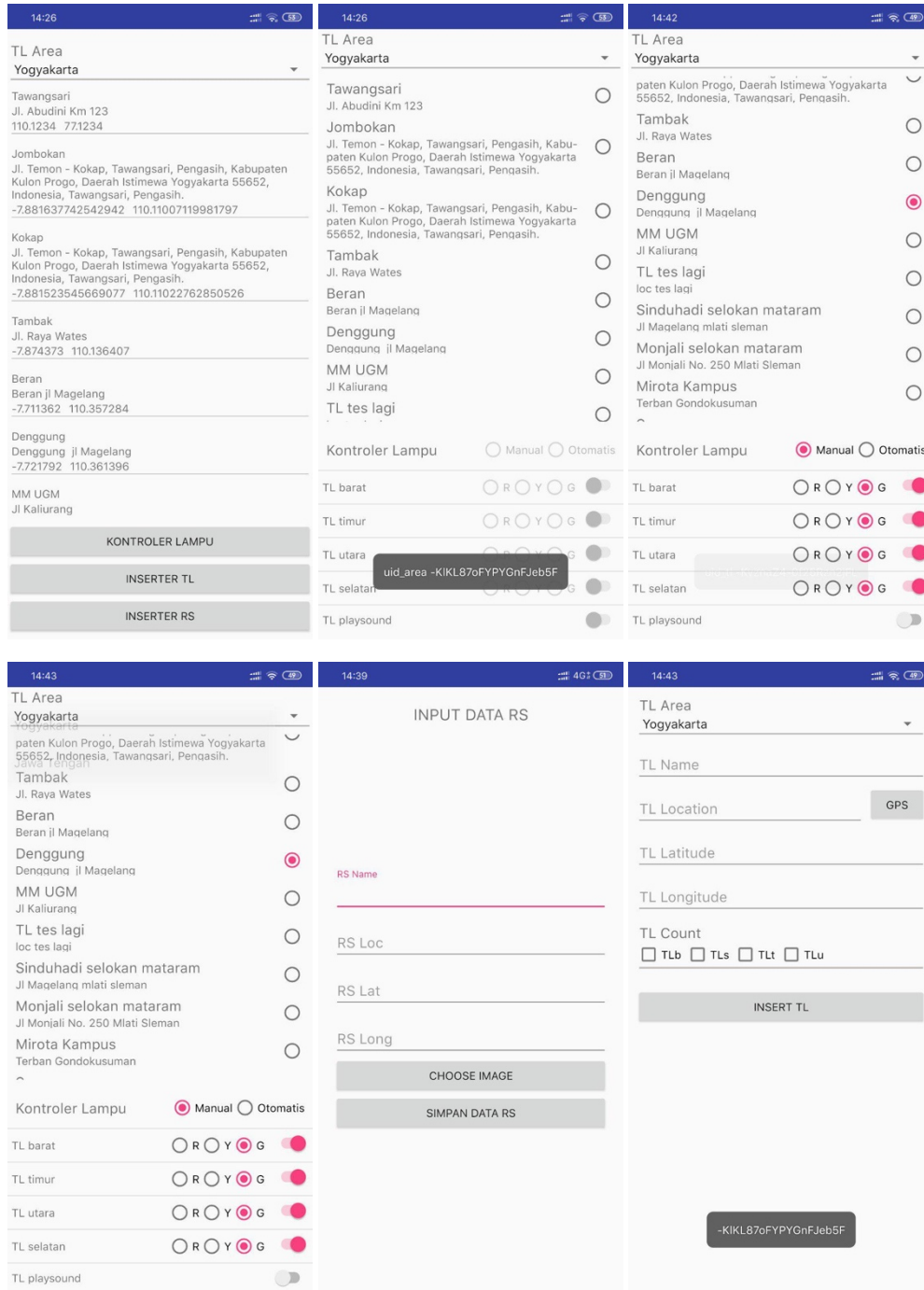
### 3) Penentuan Titik Tujuan Secara Manual

Pengguna dapat memilih titik tujuan secara manual apabila tujuan tidak tersedia di daftar. Fasilitas ini adalah fasilitas yang disediakan untuk layanan kenaraan prioritas Pemadam Kebakaran dan Pengawal kendaraan khusus. Layanan yang diberikan memberikan keleluasaan pengguna untuk mencari titik tujuan menggunakan pencarian kata ataupun menggunakan lokasi titik yang terdapat dalam peta digital.



#### 4) Halaman Admin

Halaman Admin berisikan konfigurasi data lokasi Smart Traffic Light berikut penambahan dan setting. Selain itu pada halaman ini juga bisa digunakan untuk menambahkan data lokasi GPS serta gambar dari Rumah Sakit. Data nyala lampu APILL akan terbaca melalui aplikasi dalam menu ini



```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.app.ProgressDialog;
4 import android.content.BroadcastReceiver;
5 import android.content.Context;
6 import android.content.Intent;
7 import android.content.IntentFilter;
8 import android.graphics.Color;
9 import android.location.Address;
10 import android.location.Geocoder;
11 import android.location.Location;
12 import android.location.LocationListener;
13 import android.location.LocationManager;
14 import android.os.CountDownTimer;
15 import android.support.v7.app.AppCompatActivity;
16 import android.os.Bundle;
17 import android.text.TextUtils;
18 import android.util.Log;
19 import android.view.WindowManager;
20 import android.widget.TextView;
21 import android.widget.Toast;
22
23 import com.akexorcist.googledirection.DirectionCallback;
24 import com.akexorcist.googledirection.GoogleDirection;
25 import com.akexorcist.googledirection.constant.AvoidType;
26 import com.akexorcist.googledirection.model.Direction;
27 import com.akexorcist.googledirection.model.Route;
28 import com.akexorcist.googledirection.util.
    DirectionConverter;
29 import com.google.android.gms.maps.CameraUpdateFactory;
30 import com.google.android.gms.maps.GoogleMap;
31 import com.google.android.gms.maps.OnMapReadyCallback;
32 import com.google.android.gms.maps.SupportMapFragment;
33 import com.google.android.gms.maps.model.
    BitmapDescriptorFactory;
34 import com.google.android.gms.maps.model.CameraPosition;
35 import com.google.android.gms.maps.model.LatLng;
36 import com.google.android.gms.maps.model.Marker;
37 import com.google.android.gms.maps.model.MarkerOptions;
38 import com.google.android.gms.maps.model.PolylineOptions;
39 import com.google.android.gms.tasks.Task;
40 import com.google.firebase.auth.FirebaseAuth;
41 import com.google.firebase.database.DataSnapshot;
42 import com.google.firebase.database.DatabaseError;
43 import com.google.firebase.database.DatabaseReference;
44 import com.google.firebase.database.FirebaseDatabase;
45 import com.google.firebase.database.ValueEventListener;
46 import com.google.maps.android.PolyUtil;
47 import com.google.maps.android.SphericalUtil;
48 import com.google.maps.android.ui.IconGenerator;
49
50 import java.io.IOException;
51 import java.util.ArrayList;
52 import java.util.List;
53 import java.util.Locale;
```



```
54
55 import aplikasiku.navwithdirectionlib.FDBMobel.
    LogPerjalanan;
56 import aplikasiku.navwithdirectionlib.FDBMobel.
    NewSkemaRequest;
57 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
58 import aplikasiku.navwithdirectionlib.FDBMobel.Requester;
59 import aplikasiku.navwithdirectionlib.GPS.GoogleService;
60 import aplikasiku.navwithdirectionlib.GPS.LatLngLoc;
61 import aplikasiku.navwithdirectionlib.GPS.SortTLinRoute;
62
63
64 public class hal_map extends AppCompatActivity {
65     private static final String TAG = hal_map.class.
        getName();
66
67     GoogleMap gogelMap;
68     String serverKey = "
    AIzaSyBPWWTZcnHwIfe8c_gLJkaLm8WGdJnLS7Y";
69
70     LatLng origin = new LatLng(DataPerjalanan.originLat,
    DataPerjalanan.originLong);
71     LatLng defaultViewMap = new LatLng(DataPerjalanan.
    originLat,DataPerjalanan.originLong);
72     LatLng destination = new LatLng(DataPerjalanan.
    destinationLat,DataPerjalanan.destinationLong);
73
74     ArrayList<LatLng> directionRoute;
75     ArrayList<LatLngLoc> testPoint;
76     ArrayList<LatLngLoc> TLinRoute;
77     ArrayList<LatLngLoc> sTLinRoute;
78
79     ArrayList<LatLngLoc> PointListTL;
80
81     Double latitude, longitude;
82     Geocoder geocoder;
83     SortTLinRoute sortTLinRoute;
84
85     Marker markerAmbulance;
86     Marker markerRS;
87
88     FirebaseDatabase mFirebaseInstance;
89     FirebaseAuth auth;
90     private String uid_area = "-KlKL87oFYPYGnFJeb5F";
91
92     TextView halInfo;
93
94     private int noTLonRoute =0;
95     String sound = "mati";
96
97     @Override
98     protected void onCreate(Bundle savedInstanceState) {
99         super.onCreate(savedInstanceState);
100         setContentView(R.layout.activity_hal_map);
101         getWindow().addFlags(WindowManager.LayoutParams.
```

```

101 FLAG_KEEP_SCREEN_ON);
102
103     mFirebaseInstance = FirebaseDatabase.getInstance()
104     ;
105     auth = FirebaseAuth.getInstance();
106     ((SupportMapFragment) getSupportFragmentManager().
107     findFragmentById(R.id.map)).getMapAsync(new
108     OnMapReadyCallback() {
109         @Override
110         public void onMapReady(GoogleMap googleMap) {
111             gogelMap = googleMap;
112             gogelMap.animateCamera(CameraUpdateFactory
113             .newLatLngZoom(origin, 13));
114         }
115     });
116     testPoint = new ArrayList<LatLngLoc>();
117     TLinRoute = new ArrayList<LatLngLoc>();
118
119     PointListTL = new ArrayList<LatLngLoc>();
120     sTLinRoute= new ArrayList<LatLngLoc>();
121     sTLinRoute = null;
122
123     halInfo = (TextView) findViewById(R.id.halmapinfo)
124     ;
125     //Toast.makeText(getApplicationContext(),
126     DataPerjalanan.destination, Toast.LENGTH_SHORT).show();
127     new CountDownTimer(2000,1000){
128         @Override
129         public void onTick(long millisUntilFinished) {
130
131         }
132
133         @Override
134         public void onFinish() {
135             //Toast.makeText(getApplicationContext
136             (), "Get direction", Toast.LENGTH_SHORT).show();
137             requestDirection();
138             registerReceiver(broadcastReceiver, new
139             IntentFilter(GoogleService.str_receiver));
140         }
141     }.start();
142
143
144     private void requestDirection() {
145         GoogleDirection.withServerKey(serverKey)
146             .from(origin)
147             .to(destination)

```

```

148         .optimizeWaypoints(true)
149         .alternativeRoute(true)
150         .execute(new DirectionCallback() {
151             @Override
152             public void onDirectionSuccess(
Direction direction, String rawBody) {
153                 //direction.getRouteList().get(0).
getLegList().get(0).getStepList() //titik turn by turn
154                 ArrayList<LatLng>
directionPositionList = direction.getRouteList().get(0).
getLegList().get(0).getDirectionPoint();
155                 directionRoute =
directionPositionList;
156                 gogelMap.addPolyline(
DirectionConverter.createPolyline(getApplicationContext(),
directionPositionList, 8, Color.BLUE));
157                 gogelMap.setTrafficEnabled(true);
158                 gogelMap.getUiSettings().
setZoomControlsEnabled(true);
159
160                 // Enable / Disable my location
button
161                 gogelMap.getUiSettings().
setMyLocationButtonEnabled(true);
162
163                 // Enable / Disable Compass icon
164                 gogelMap.getUiSettings().
setCompassEnabled(true);
165
166                 // Enable / Disable Rotate gesture
167                 gogelMap.getUiSettings().
setRotateGesturesEnabled(true);
168
169                 // Enable / Disable zooming
functionality
170                 gogelMap.getUiSettings().
setZoomGesturesEnabled(true);
171                 addMarkerAmbulance(origin, "Start")
;
172                 addMarkerRS(destination,
DataPerjalanan.destination);
173
174                 firebaseGetDataTL(uid_area);
175                 /*
176                 //demo compute heading//
177                 LatLngLoc tugu = new LatLngLoc(new
LatLng(-7.782884, 110.367098), "TUGU JOGJA");
178                 LatLngLoc bkd = new LatLngLoc(new
LatLng(-7.782027, 110.356351), "BKD JOGJA");
179                 LatLngLoc smk3yk = new LatLngLoc(
new LatLng(-7.777586, 110.366027), "SMK3YK");
180                 LatLngLoc mcdonald = new LatLngLoc
(new LatLng(-7.783260, 110.371703), "McDONALD");
181                 LatLngLoc PLNmangkubumi = new
LatLngLoc(new LatLng(-7.787897, 110.366830), "

```

```

181 PLNmangkubumi");
182
183         gogelMap.addMarker(new
MarkerOptions().position(tugu.getLocation()).title(tugu.
getNameLoc()));
184         gogelMap.addMarker(new
MarkerOptions().position(bkd.getLocation()).title(bkd.
getNameLoc() + ": " + arahAsalAmbulance(tugu.getLocation
(), bkd.getLocation()))).showInfoWindow();
185         gogelMap.addMarker(new
MarkerOptions().position(smk3yk.getLocation()).title(
smk3yk.getNameLoc() + ": " + arahAsalAmbulance(tugu.
getLocation(), smk3yk.getLocation()));
186         gogelMap.addMarker(new
MarkerOptions().position(mcdonald.getLocation()).title(
mcdonald.getNameLoc() + ": " + arahAsalAmbulance(tugu.
getLocation(), mcdonald.getLocation()));
187         gogelMap.addMarker(new
MarkerOptions().position(PLNmangkubumi.getLocation()).
title(PLNmangkubumi.getNameLoc() + ": " +
arahAsalAmbulance(tugu.getLocation(), PLNmangkubumi.
getLocation())));
188
189         //demo update tl
190         updateRequestTL("-
KvgGfURCFOLpqV_60E4", "TLu", 50);
191         */
192     }
193
194     @Override
195     public void onDirectionFailure(
Throwable t) {
196
197     }
198     });
199 }
200
201     private void firebaseGetDataTL(String uid_area) {
202         mFirebaseInstance.getReference("new-tl").child(
uid_area).addListenerForSingleValueEvent(new
 ValueEventListener() {
203             @Override
204             public void onDataChange(DataSnapshot
dataSnapshot) {
205                 setTLonRoute(dataSnapshot);
206             }
207
208             @Override
209             public void onCancelled(DatabaseError
databaseError) {
210
211             }
212         });
213     }
214

```

```

215     private void setTLonRoute(DataSnapshot dataSnapshot) {
216         //Toast.makeText(getApplicationContext(), "
SetTLonRoute", Toast.LENGTH_SHORT).show();
217         NewSkemaTL[] tldata = new NewSkemaTL[(int)
dataSnapshot.getChildrenCount()];
218         int count = 0;
219         for(DataSnapshot data : dataSnapshot.getChildren()
){
220             NewSkemaTL tl = data.getValue(NewSkemaTL.class
);
221             LatLng tlLatLng = new LatLng(tl.TLlat, tl.
TLlong);
222             PointListTL.add(new LatLngLoc(tlLatLng, tl.
TLname, data.getKey()));
223             tldata[count] = tl;
224             count++;
225         }
226
227         for (int i = 0; i < PointListTL.size(); i++){
228             Boolean dijalur = PolyUtil.isLocationOnPath(
PointListTL.get(i).getLocation(),directionRoute, false,10);
229             if (dijalur) {
230                 TLinRoute.add(PointListTL.get(i));
231                 addMarkerTL(PointListTL.get(i).getLocation
());
232                 //Toast.makeText(getApplicationContext(),
tldata[i].TLname + " on route", Toast.LENGTH_SHORT).show
();
233             }
234             else{
235                 //Toast.makeText(getApplicationContext(),
tldata[i].TLname + " not on route", Toast.LENGTH_SHORT).
show();
236             }
237         }
238
239         sortTLinRoute = new SortTLinRoute(origin,TLinRoute
);
240         sTLinRoute = sortTLinRoute.getSortTLInRoute();
241
242         LatLngLoc RS = new LatLngLoc(destination,
DataPerjalanan.destination);
243         sTLinRoute.add(RS);
244
245         for (int z = 0; z < sTLinRoute.size(); z++){
246             Log.d(TAG, "onDirectionSuccess: "+ z +" nama "
+ sTLinRoute.get(z).getNameLoc() );
247         }
248     }
249
250     private void setNewRouting() {
251         final DatabaseReference mDbRoute;
252         mDbRoute = mFirebaseInstance.getReference("routing
");
253

```

```

254         final String uid_route = mDbRoute.push().getKey();
255         mDbRoute.child(uid_route).setValue(new aplikasiku.
navwithdirectionlib.FDBMobel.Route(auth.getCurrentUser().
getUid(), "-keytl1,-keytl2,keytl3", new LogPerjalanan()),
new DatabaseReference.CompletionListener() {
256             @Override
257             public void onComplete(DatabaseError
databaseError, DatabaseReference databaseReference) {
258                 if(databaseError==null){
259                     //Toast.makeText(getApplicationContext
()), "Insert route oke", Toast.LENGTH_SHORT).show();
260                     mDbRoute.child(uid_route).child("log")
.push().setValue(new LogPerjalanan("alamatstart", 77.0,
110.0), new DatabaseReference.CompletionListener() {
261                         @Override
262                         public void onComplete(
DatabaseError databaseError, DatabaseReference
databaseReference) {
263                             if(databaseError==null){
264                                 //Toast.makeText(
getApplicationContext(), "Insert log oke", Toast.
LENGTH_SHORT).show();
265                                 }
266                             }
267                         });
268                     }
269                 }
270             });
271         }
272
273         /**
274          * Get data from service GoogleService
275          */
276
277         @Override
278         protected void onResume() {
279             super.onResume();
280             registerReceiver(broadcastReceiver, new
IntentFilter(GoogleService.str_receiver));
281         }
282
283         @Override
284         protected void onPause() {
285             super.onPause();
286             unregisterReceiver(broadcastReceiver);
287         }
288
289         private String formatNumber(double distance) {
290             String unit = "m";
291             if (distance < 1) {
292                 distance *= 1000;
293                 unit = "mm";
294             } else if (distance > 1000) {
295                 distance /= 1000;
296                 unit = "km";

```

```
297     }
298     return String.format("%4.3f%s", distance, unit);
299 }
300
301 private void addMarkerRS(LatLng position,String nmRS)
302 {
303     IconGenerator iconFactory = new IconGenerator(this
304 );
305     MarkerOptions markerOptions = new MarkerOptions().
306     icon(BitmapDescriptorFactory.fromResource(
307 R.drawable.marker_rs)).
308     position(position).title(nmRS).
309     anchor(iconFactory.getAnchorU(),
310 iconFactory.getAnchorV()).zIndex(10);
311
312     markerRS = gogelMap.addMarker(markerOptions);
313     markerRS.showInfoWindow();
314 }
315
316 private void addMarkerAmbulance(LatLng position,String
317 jarak) {
318     IconGenerator iconFactory = new IconGenerator(this
319 );
320     MarkerOptions markerOptions = new MarkerOptions().
321     icon(BitmapDescriptorFactory.fromResource(
322 R.drawable.marker_ambulance)).
323     position(position).title(jarak).visible(
324 true).
325     anchor(iconFactory.getAnchorU(),
326 iconFactory.getAnchorV()).zIndex(11);
327     markerAmbulance = gogelMap.addMarker(markerOptions
328 );
329     markerAmbulance.showInfoWindow();
330 }
331
332 private void addMarkerTL(LatLng position) {
333
334     IconGenerator iconFactory = new IconGenerator(this
335 );
336     MarkerOptions markerOptions = new MarkerOptions().
337     icon(BitmapDescriptorFactory.fromResource(
338 R.drawable.marker_tl)).
339     position(position).
340     anchor(iconFactory.getAnchorU(),
341 iconFactory.getAnchorV()).zIndex(9);
342     gogelMap.addMarker(markerOptions);
343 }
344
345 private void updateMarkerAmbulance(LatLng locNow,
346 String jarakTL,String nmTL) {
347     //markerAmbulance.remove();
348     if (sTLinRoute != null) {
349         markerAmbulance.setPosition(locNow);
350         markerAmbulance.setTitle(jarakTL);
351         markerAmbulance.setSnippet("Arah ke TL "+nmTL)
```

```

337 ;
338         markerAmbulance.showInfoWindow();
339
340 //             CameraPosition cameraPosition = new
CameraPosition.Builder().target(
341 //                 new LatLng(locNow.latitude, locNow.
longitude)).zoom(12).build();
342 //
343 //             gogelMap.animateCamera(CameraUpdateFactory.
newCameraPosition(cameraPosition));
344     }
345     //addMarkerAmbulance(locNow, jarakTL);
346 }
347
348 /**
349  * Get data from service GoogleService
350  */
351
352     private BroadcastReceiver broadcastReceiver = new
BroadcastReceiver() {
353         @Override
354         public void onReceive(Context context, Intent
intent) {
355
356             latitude = Double.valueOf(intent.
getStringExtra("latutide"));
357             longitude = Double.valueOf(intent.
getStringExtra("longitude"));
358             LatLng LocNow = new LatLng(latitude, longitude
);
359             if (sTLinRoute != null) {
360                 double distance = SphericalUtil.
computeDistanceBetween(LocNow, sTLinRoute.get(noTLonRoute)
.getLocation());
361
362                 //gogelMap.animateCamera(
CameraUpdateFactory.newLatLngZoom(LocNow, 13));
363
364                 LatLng area1 = new LatLng(-7.768131, 110.
387519);
365                 LatLng area2 = new LatLng(-7.767976, 110.
387569);
366                 LatLng area3 = new LatLng(-7.768028, 110.
387677);
367                 LatLng area4 = new LatLng(-7.768163, 110.
387625);
368
369                 ArrayList<LatLng> polyArea = new ArrayList
<LatLng>();
370                 polyArea.add(area1);
371                 polyArea.add(area2);
372                 polyArea.add(area3);
373                 polyArea.add(area4);
374
375                 String Jarak = formatNumber(distance);

```



```

376
377 //          if ((PolyUtil.isLocationOnEdge(LocNow,
polyArea,false,200))) {
378 //          noTLonRoute++;
379 //          sound = "Mati";
380 //          }
381 int jmlpoint = sTLinRoute.size();
382 if (distance < 120){
383     if (noTLonRoute < jmlpoint - 1) {
384         noTLonRoute++;
385         sound = "matii";
386     }
387 }
388 if ((distance < 800) && (distance>=120)){
389     sound = "bunyi";
390
391     //update tl and add request
392     if(!TextUtils.isEmpty(sTLinRoute.get(
noTLonRoute).getTLkey())){
393         String tlkey = sTLinRoute.get(
noTLonRoute).getTLkey();
394         String TLbtus = arahAsalAmbulance(
sTLinRoute.get(noTLonRoute).getLocation(),LocNow);
395         updateRequestTL(tlkey, TLbtus,
distance);
396     }
397 }
398
399 String text = new String(
400     "Lokasi Lat " + latitude + " Lng "
+ longitude
401     + " ID TL "+ noTLonRoute
402     + " Jarak dg TL "+
sTLinRoute.get(noTLonRoute).getNameLoc()
403     + " adl " + Jarak
404     + " Sound " + sound);
405
406     halInfo.setText(text);
407     updateMarkerAmbulance(LocNow,Jarak,
sTLinRoute.get(noTLonRoute).getNameLoc());
408 } else {
409     Toast.makeText(getApplicationContext(), "
Datanya sTLinRoute Kosong", Toast.LENGTH_SHORT).show();
410 }
411 }
412 };
413 };
414
415 private String arahAsalAmbulance(LatLng tldes, LatLng
ambulance){
416     String arah = "";
417     double derajat = SphericalUtil.computeHeading(
tldes, ambulance);
418     if(derajat>=-45 && derajat <=45){
419         arah = "TLu";

```

```

420     }
421     else if(derajat>45 && derajat <=135){
422         arah = "TLt";
423     }
424     else if(derajat>135 || derajat<-135){
425         arah = "TLs";
426     }
427     else if(derajat<-45 && derajat>=-135){
428         arah = "TLb";
429     }
430
431     Log.d("DERAJAT", Double.toString(derajat) + " " +
    arah);
432     return arah;
433 }
434
435 private void updateRequestTL(final String keyTL, final
    String TLbtus, final double distance){
436     mFirebaseInstance.getReference("new-tl").child(
    uid_area).child(keyTL).child("TLpos").child(TLbtus).
    addListenerForSingleValueEvent(new ValueEventListener() {
437         @Override
438         public void onDataChange(DataSnapshot
    dataSnapshot) {
439             if(dataSnapshot.exists()){
440                 mFirebaseInstance.getReference("new-tl
    ").child(uid_area).child(keyTL).child("TLpos").child(
    TLbtus).child("request").child("perequest").
    addListenerForSingleValueEvent(new ValueEventListener() {
441                     @Override
442                     public void onDataChange(final
    DataSnapshot dataSnapshot1) {
443                         if(!dataSnapshot1.exists()){
444                             updateTL(keyTL, TLbtus,
    distance, 1);
445                             mFirebaseInstance.
    getReference("new-tl").child(uid_area).child(keyTL).child(
    "TLpos").child(TLbtus).child("request").child("useNumber")
    .setValue(1);
446                         }
447                         else{
448                             mFirebaseInstance.
    getReference("new-tl").child(uid_area).child(keyTL).child(
    "TLpos").child(TLbtus).child("request").child("perequest")
    .child(auth.getCurrentUser().getUid()).
    addListenerForSingleValueEvent(new ValueEventListener() {
449                                 @Override
450                                 public void
    onDataChange(DataSnapshot dataSnapshot2) {
451                                     if(dataSnapshot2.
    exists()){
452                                         final
    Requester requester = dataSnapshot2.getValue(Requester.
    class);
453

```

```

453 mFirebaseInstance.getReference("new-tl").child(uid_area).
    child(keyTL).child("TLpos").child(TLbtus).child("request")
454     .addListenerForSingleValueEvent(new ValueEventListener() {
455         @Override
456         public
457         void onDataChange(DataSnapshot dataSnapshot3) {
458             NewSkemaRequest req = dataSnapshot3.getValue(
459                 NewSkemaRequest.class);
460             if(
461             requester.antrian==req.useNumber){
462                 updateTL(keyTL, TLbtus, distance, requester.antrian);
463             }
464         }
465     });
466     }
467     @Override
468     public
469     void onCancelled(DatabaseError databaseError) {
470         }
471     }
472     }
473     }
474     @Override
475     public void
476     onCancelled(DatabaseError databaseError) {
477         }
478     }
479     }
480     }
481     }
482     @Override
483     public void onCancelled(
484     DatabaseError databaseError) {
485         }
486     }
487     });
488     }
489     else{
490         //tidak ada tl point dari arah
         ambulance yg di dapat
    }

```

```
491     }
492
493     @Override
494     public void onCancelled(DatabaseError
databaseError) {
495
496     }
497     });
498 }
499
500     //only call in updateRequestTL
501     private void updateTL(String keyTL, String TLbtus,
double distance, int antrian) {
502         mFirebaseInstance.getReference("new-tl").child(
uid_area).child(keyTL).child("TLpos").child(TLbtus).child(
"request").child("perequest").child(auth.getCurrentUser().
getUid()).setValue(new Requester("ambulance", antrian));
503         mFirebaseInstance.getReference("new-tl").child(
uid_area).child(keyTL).child("TLpos").child(TLbtus).child(
"request").child("adaRequest").setValue(true);
504         mFirebaseInstance.getReference("new-tl").child(
uid_area).child(keyTL).child("TLpos").child(TLbtus).child(
"request").child("jarakAcuan").setValue(distance);
505         mFirebaseInstance.getReference("new-tl").child(
uid_area).child(keyTL).child("TLpos").child(TLbtus).child(
"isEmergency").setValue(true);
506         mFirebaseInstance.getReference("new-tl").child(
uid_area).child(keyTL).child("TLplaysound").setValue(true)
;
507     }
508
509     private void removeRequestTL(){
510
511     }
512 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3
4 import android.app.ActivityManager;
5 import android.content.BroadcastReceiver;
6 import android.content.Context;
7 import android.content.Intent;
8 import android.content.IntentFilter;
9 import android.content.SharedPreferences;
10 import android.content.pm.PackageManager;
11 import android.location.Address;
12 import android.location.Geocoder;
13 import android.preference.PreferenceManager;
14 import android.support.v4.app.ActivityCompat;
15 import android.support.v4.content.ContextCompat;
16 import android.support.v7.app.AppCompatActivity;
17 import android.os.Bundle;
18 import android.support.v7.widget.LinearLayoutManager;
19 import android.support.v7.widget.RecyclerView;
20 import android.view.View;
21 import android.widget.Button;
22 import android.widget.TextView;
23 import android.widget.Toast;
24
25 import com.google.firebase.auth.FirebaseAuth;
26 import com.google.firebase.database.DataSnapshot;
27 import com.google.firebase.database.DatabaseError;
28 import com.google.firebase.database.DatabaseReference;
29 import com.google.firebase.database.FirebaseDatabase;
30 import com.google.firebase.database.ValueEventListener;
31
32 import java.io.IOException;
33 import java.util.ArrayList;
34 import java.util.List;
35 import java.util.Locale;
36
37 import aplikasiku.navwithdirectionlib.FDBMobel.Hospital;
38 import aplikasiku.navwithdirectionlib.GPS.GoogleService;
39
40 public class hal_home extends AppCompatActivity {
41
42     private RecyclerView mRecyclerView;
43     private RecyclerView.Adapter mAdapter;
44     private RecyclerView.LayoutManager mLayoutManager;
45
46     FirebaseAuth auth;
47     TextView tvIdLogin, tvPosisiLogin, tvLatLogin,
48     tvLongLogin;
49     Button butKontrolerLampu, butSetLampu;
50
51     private static final int REQUEST_PERMISSIONS = 100;
52     boolean boolean_permission;
53     SharedPreferences mPref;
54     SharedPreferences.Editor medit;
```

```
55     Double latitude, longitude;
56     Geocoder geocoder;
57
58     // Creating DatabaseReference.
59     DatabaseReference databaseReference;
60
61     // Creating List of ImageUploadInfo class.
62     List<Hospital> list = new ArrayList<>();
63
64     @Override
65     protected void onCreate(Bundle savedInstanceState) {
66         super.onCreate(savedInstanceState);
67         setContentView(R.layout.activity_hal_home);
68
69         init();
70         getDataLogin();
71         getLocGPS();
72     }
73
74     private BroadcastReceiver broadcastReceiver = new
BroadcastReceiver() {
75         @Override
76         public void onReceive(Context context, Intent
intent) {
77
78             latitude = Double.valueOf(intent.
getStringExtra("latitude"));
79             longitude = Double.valueOf(intent.
getStringExtra("longitude"));
80
81             List<Address> addresses = null;
82
83             try {
84                 addresses = geocoder.getFromLocation(
latitude, longitude, 1);
85
86                 if(addresses.size()>0){
87                     String gpsJalan = addresses.get(0).
getAddressLine(0);
88                     String gpsDesa = addresses.get(0).
getSubLocality();
89                     String gpsKecamatan = addresses.get(0)
.getLocality();
90                     String gpsKabupaten = addresses.get(0)
.getSubAdminArea();
91                     String gpsProvinsi = addresses.get(0).
getAdminArea();
92                     String gpsNegara = addresses.get(0).
getCountryName();
93                     String alamat = gpsJalan + ", " +
gpsDesa + ", " + gpsKecamatan + ". ";
94                     //alamat+= gpsKabupaten + ", " +
gpsProvinsi + ", " + gpsNegara;
95
96                     tvPosisiLogin.setText(alamat);
```

```

97         tvLatLogin.setText("Lat: " + Double.
    toString(latitude));
98         tvLongLogin.setText("Long: " + Double.
    toString(longitude));
99
100         //to set on next intent
101         DataPerjalanan.originLat = latitude;
102         DataPerjalanan.originLong = longitude;
103     }
104
105     } catch (IOException e1) {
106         e1.printStackTrace();
107     }
108 }
109 };
110
111 private void getLocGPS(){
112     if (boolean_permission) {
113         if (isMyServiceRunning(GoogleService.class)){
114             Toast.makeText(getApplicationContext(), "
Service sudah running", Toast.LENGTH_SHORT).show();
115         } else {
116             Toast.makeText(getApplicationContext(), "
Service belum running", Toast.LENGTH_SHORT).show();
117             Intent intent = new Intent(
    getApplicationContext(), GoogleService.class);
118             startService(intent);
119             registerReceiver(broadcastReceiver, new
    IntentFilter(GoogleService.str_receiver));
120         }
121
122
123     } else {
124         Toast.makeText(getApplicationContext(), "
Please enable the gps", Toast.LENGTH_SHORT).show();
125     }
126 }
127
128 private void getDataLogin() {
129     auth = FirebaseAuth.getInstance();
130     if (auth.getCurrentUser() != null) {
131         tvIdLogin.setText(auth.getCurrentUser().
    getEmail());
132         Toast.makeText(getApplicationContext(), "Get
    data login", Toast.LENGTH_SHORT).show();
133     }
134 }
135
136 private void init() {
137     tvIdLogin = (TextView) findViewById(R.id.
    textViewIdLogin);
138     tvPosisiLogin = (TextView) findViewById(R.id.
    textViewPosisiLogin);
139     tvLatLogin = (TextView) findViewById(R.id.
    textViewLatLogin);

```

```
140         tvLongLogin = (TextView) findViewById(R.id.
    textViewLongLogin);
141         butKontrolerLampu = (Button) findViewById(R.id.
    buttonKontrolerLampu);
142
143         butSetLampu = (Button) findViewById(R.id.
    butSetDestination);
144         butSetLampu.setOnClickListener(new View.
    OnClickListener() {
145             @Override
146             public void onClick(View v) {
147                 if(tvPosisiLogin.getText().toString().
    equals("posisi")){
148                     Toast.makeText(getApplicationContext()
    , "Masih mencari posisi GPS", Toast.LENGTH_SHORT).show();
149                 }
150                 else{
151                     startActivity(new Intent(hal_home.this
    , SetDestinationActivity.class)
152                         .putExtra("origin_address",
    tvPosisiLogin.getText().toString()));
153                 }
154             }
155         });
156
157         mRecyclerView = (RecyclerView) findViewById(R.id.
    recyclerViewListRS);
158         mRecyclerView.setHasFixedSize(true);
159
160         mLayoutManager = new LinearLayoutManager(this);
161         mRecyclerView.setLayoutManager(mLayoutManager);
162
163         /**
164          *
165          */
166
167         geocoder = new Geocoder(this, Locale.getDefault())
    ;
168         mPref = PreferenceManager.
    getDefaultSharedPreferences(getApplicationContext());
169         medit = mPref.edit();
170
171         butKontrolerLampu.setOnClickListener(new View.
    OnClickListener() {
172             @Override
173             public void onClick(View v) {
174                 startActivity(new Intent(hal_home.this,
    AdminActivity.class));
175             }
176         });
177
178         databaseReference = FirebaseDatabase.getInstance()
    .getReference(InsertRsFirebase.Database_Path);
179
180
```



```
181         databaseReference.addValueEventListener(new
ValueEventListener() {
182             @Override
183             public void onDataChange(DataSnapshot snapshot
) {
184
185                 for (DataSnapshot postSnapshot : snapshot.
getChildren()) {
186
187                     Hospital hospital = postSnapshot.
getValue(Hospital.class);
188
189                     list.add(hospital);
190                 }
191
192                 mAdapter = new RSRecyclerViewAdapter(
getApplicationContext(), list, new RSRecyclerViewAdapter.
CustomItemClickListener() {
193                     @Override
194                     public void onItemClick(View v, int
position) {
195
196                         if((DataPerjalanan.originLat==0)
|| (DataPerjalanan.originLong==0)){
197                             Toast.makeText(
getApplicationContext(), "Latitude longitude user masih
dicari", Toast.LENGTH_SHORT).show();
198                         }
199                         else{
200                             DataPerjalanan.destination =
list.get(position).RsName;
201                             DataPerjalanan.destinationLat
= list.get(position).RsLat;
202                             DataPerjalanan.destinationLong
= list.get(position).RsLong;
203
204                             Intent intentMap = new Intent(
hal_home.this, hal_map.class);
205                             startActivity(intentMap);
206                         }
207                     }
208                 });
209                 mRecyclerView.setAdapter(mAdapter);
210
211             }
212
213
214
215             @Override
216             public void onCancelled(DatabaseError
databaseError) {
217
218                 // Hiding the progress dialog.
219                 // progressDialog.dismiss();
220
```

```

221     }
222     });
223
224 //
225 //     final Hospital[] rsData = new Hospital[3];
226 //     rsData[0] = new Hospital();
227 //     rsData[0].RsName = "RSUP Dr. Sardjito";
228 //     rsData[0].RsLoc = "Jl. Kesehatan No. 1, Sekip,
    Sinduadi, Sleman, Sinduadi, Mlati, Kabupaten Sleman, DIY";
229 //     rsData[0].RsLat = -7.768459;
230 //     rsData[0].RsLong = 110.373480;
231 //
232 //     rsData[1] = new Hospital();
233 //     rsData[1].RsName = "RS Panti Rapih";
234 //     rsData[1].RsLoc = "Jl. Cik Di Tiro No.30,
    Caturtunggal, Kec. Depok, Kabupaten Sleman, DIY";
235 //     rsData[1].RsLat = -7.776956;
236 //     rsData[1].RsLong = 110.376149;
237 //
238 //     rsData[2] = new Hospital();
239 //     rsData[2].RsName = "RS Bethesda";
240 //     rsData[2].RsLoc = "Jl. Jend. Sudirman No.70,
    Kotabaru, Gondokusuman, Kota Yogyakarta, DIY";
241 //     rsData[2].RsLat = -7.783235;
242 //     rsData[2].RsLong = 110.378552;
243
244
245
246     fn_permission();
247 }
248
249     private void fn_permission() {
250         if ((ContextCompat.checkSelfPermission(
    getApplicationContext(), android.Manifest.permission.
    ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
    )) {
251             if ((ActivityCompat.
    shouldShowRequestPermissionRationale(hal_home.this,
    android.Manifest.permission.ACCESS_FINE_LOCATION))) {}
252             else {
253                 ActivityCompat.requestPermissions(hal_home
    .this, new String[]{android.Manifest.permission.
    ACCESS_FINE_LOCATION}, REQUEST_PERMISSIONS);
254             }
255             } else {
256                 boolean_permission = true;
257             }
258         }
259
260     @Override
261     protected void onResume() {
262         super.onResume();
263         registerReceiver(broadcastReceiver, new
    IntentFilter(GoogleService.str_receiver));
264     }

```

```
265
266     @Override
267     protected void onPause() {
268         super.onPause();
269         unregisterReceiver(broadcastReceiver);
270     }
271
272     private boolean isMyServiceRunning(Class<?>
serviceClass) {
273         ActivityManager manager = (ActivityManager)
getService(Context.ACTIVITY_SERVICE);
274         for (ActivityManager.RunningServiceInfo service :
manager.getRunningServices(Integer.MAX_VALUE)) {
275             if (serviceClass.getName().equals(service.
service.getClassName())) {
276                 return true;
277             }
278         }
279         return false;
280     }
281 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.support.v7.widget.LinearLayoutManager;
7 import android.support.v7.widget.RecyclerView;
8 import android.view.View;
9 import android.widget.AdapterView;
10 import android.widget.Button;
11 import android.widget.Spinner;
12 import android.widget.Toast;
13
14 import com.google.firebase.database.DataSnapshot;
15 import com.google.firebase.database.DatabaseError;
16 import com.google.firebase.database.FirebaseDatabase;
17 import com.google.firebase.database.ValueEventListener;
18
19 import aplikasiku.navwithdirectionlib.FDBMobel.Area;
20 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
21
22 public class AdminActivity extends AppCompatActivity {
23
24     Button butKontrolerLampu, butInsertTL, butInsertRS;
25     Spinner spinnerAdminTLArea;
26     RecyclerView recyclerViewTLlist;
27     FirebaseDatabase database;
28
29     String uid_area;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_admin);
35
36         database = FirebaseDatabase.getInstance();
37
38         butKontrolerLampu = (Button) findViewById(R.id.
buttonAdminKontrolLampu);
39         butInsertTL = (Button) findViewById(R.id.
buttonAdminInsertTL);
40         butInsertRS = (Button) findViewById(R.id.
buttonAdminInsertRS);
41
42         spinnerAdminTLArea = (Spinner) findViewById(R.id.
spinnerAdminTLArea);
43         recyclerViewTLlist = (RecyclerView) findViewById(R.
id.recyclerview_admin_tllist);
44         recyclerViewTLlist.setHasFixedSize(true);
45         LinearLayoutManager layoutManager = new
LinearLayoutManager(this);
46         recyclerViewTLlist.setLayoutManager(layoutManager);
47
48         butKontrolerLampu.setOnClickListener(new View.
OnClickListener() {
```

```

49         @Override
50         public void onClick(View v) {
51             startActivity(new Intent(AdminActivity.
this, KontrolerLampuActivity.class));
52         }
53     });
54
55     butInsertTL.setOnClickListener(new View.
OnClickListener() {
56         @Override
57         public void onClick(View v) {
58             startActivity(new Intent(AdminActivity.
this, SkemaTLbaruActivity.class));
59         }
60     });
61
62     butInsertRS.setOnClickListener(new View.
OnClickListener() {
63         @Override
64         public void onClick(View v) {
65             startActivity(new Intent(AdminActivity.
this, InsertRsFirebase.class));
66         }
67     });
68
69     spinnerAdminTLArea.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
70         @Override
71         public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {
72             database.getReference("area").
addListenerForSingleValueEvent(new ValueEventListener() {
73                 @Override
74                 public void onDataChange(DataSnapshot
dataSnapshot) {
75                     for(DataSnapshot area :
dataSnapshot.getChildren()){
76                         Area x = area.getValue(Area.
class);
77                         if(x.area_name.equals(
spinnerAdminTLArea.getSelectedItem().toString())){
78                             uid_area = area.getKey();
79                             initGetListTL();
80                             Toast.makeText(
getApplicationContext(), "uid_area " + uid_area, Toast.
LENGTH_SHORT).show();
81                     }
82                 }
83             }
84
85         @Override
86         public void onCancelled(DatabaseError
databaseError) {
87
88         }

```

```
89         });
90     }
91
92     @Override
93     public void onNothingSelected(AdapterView<?>
parent) {
94
95     }
96     });
97 }
98
99     private void initGetListTL() {
100         database.getReference("new-tl").child(uid_area).
addListenerForSingleValueEvent(new ValueEventListener() {
101             @Override
102             public void onDataChange(DataSnapshot
dataSnapshot) {
103                 NewSkemaTL[] dataTL = new NewSkemaTL[(int)
dataSnapshot.getChildrenCount()];
104                 int count=0;
105                 for(DataSnapshot data : dataSnapshot.
getChildren()){
106                     NewSkemaTL tl = data.getValue(
NewSkemaTL.class);
107                     dataTL[count] = tl;
108                     count++;
109                 }
110
111                 AdminListTLAdapter adapter = new
AdminListTLAdapter(dataTL);
112                 recyclerViewTLlist.setAdapter(adapter);
113             }
114
115             @Override
116             public void onCancelled(DatabaseError
databaseError) {
117
118             }
119         });
120     }
121 }
122
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.animation.Animator;
4 import android.animation.AnimatorListenerAdapter;
5 import android.annotation.TargetApi;
6 import android.content.Intent;
7 import android.content.pm.PackageManager;
8 import android.support.annotation.NonNull;
9 import android.support.design.widget.Snackbar;
10 import android.support.v7.app.AppCompatActivity;
11 import android.app.LoaderManager.LoaderCallbacks;
12
13 import android.content.CursorLoader;
14 import android.content.Loader;
15 import android.database.Cursor;
16 import android.net.Uri;
17 import android.os.AsyncTask;
18
19 import android.os.Build;
20 import android.os.Bundle;
21 import android.provider.ContactsContract;
22 import android.text.TextUtils;
23 import android.view.KeyEvent;
24 import android.view.View;
25 import android.view.View.OnClickListener;
26 import android.view.inputmethod.EditorInfo;
27 import android.widget.ArrayAdapter;
28 import android.widget.AutoCompleteTextView;
29 import android.widget.Button;
30 import android.widget.EditText;
31 import android.widget.TextView;
32 import android.widget.Toast;
33
34 import com.google.android.gms.tasks.OnCompleteListener;
35 import com.google.android.gms.tasks.Task;
36 import com.google.firebase.auth.AuthResult;
37 import com.google.firebase.auth.FirebaseAuth;
38
39 import java.util.ArrayList;
40 import java.util.List;
41
42 import static android.Manifest.permission.READ_CONTACTS;
43
44 /**
45  * A login screen that offers login via email/password.
46  */
47 public class LoginActivity extends AppCompatActivity
48     implements LoaderCallbacks<Cursor> {
49     /**
50      * Id to identity READ_CONTACTS permission request.
51      */
52     private static final int REQUEST_READ_CONTACTS = 0;
53
54     /**
```

```
55     * A dummy authentication store containing known user
56     * TODO: remove after connecting to a real
57     */
58     private static final String[] DUMMY_CREDENTIALS = new
String[]{
59         "foo@example.com:hello", "bar@example.com:
60         world"
61     };
62     /**
63     * Keep track of the login task to ensure we can
64     * cancel it if requested.
65     */
66     // UI references.
67     private autoCompleteTextView mEmailView;
68     private EditText mPasswordView;
69     private View mProgressView;
70     private View mLoginFormView;
71     private FirebaseAuth auth;
72
73     @Override
74     protected void onCreate(Bundle savedInstanceState) {
75         super.onCreate(savedInstanceState);
76         setContentView(R.layout.activity_login);
77
78         auth = FirebaseAuth.getInstance();
79         if(auth.getCurrentUser()!=null){
80             Toast.makeText(getApplicationContext(), "You
81             are logged in", Toast.LENGTH_SHORT).show();
82             startActivity(new Intent(LoginActivity.this,
83             hal_home.class));
84             finish();
85         }
86         // Set up the login form.
87         mEmailView = (AutoCompleteTextView) findViewById(R
88         .id.email);
89         populateAutoComplete();
90         mPasswordView = (EditText) findViewById(R.id.
91         password);
92         Button mEmailSignInButton = (Button) findViewById(
93         R.id.email_sign_in_button);
94         mEmailSignInButton.setOnClickListener(new
95         OnClickListener() {
96             @Override
97             public void onClick(View view) {
98                 attemptLogin();
99             }
100         });
101         mLoginFormView = findViewById(R.id.login_form);
```



```

99     mProgressBar = findViewById(R.id.login_progress);
100     tvSignup = (TextView) findViewById(R.id.
textviewSignup);
101     tvSignup.setOnClickListener(new OnClickListener()
{
102         @Override
103         public void onClick(View v) {
104             startActivity(new Intent(LoginActivity.
this, SignupActivity.class));
105         }
106     });
107 }
108
109 private void populateAutoComplete() {
110     if (!mayRequestContacts()) {
111         return;
112     }
113
114     getLoaderManager().initLoader(0, null, this);
115 }
116
117 private boolean mayRequestContacts() {
118     if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M)
{
119         return true;
120     }
121     if (checkSelfPermission(READ_CONTACTS) ==
PackageManager.PERMISSION_GRANTED) {
122         return true;
123     }
124     if (shouldShowRequestPermissionRationale(
READ_CONTACTS)) {
125         Snackbar.make(mEmailView, R.string.
permission_rationale, Snackbar.LENGTH_INDEFINITE)
126             .setAction(android.R.string.ok, new
View.OnClickListener() {
127                 @Override
128                 @TargetApi(Build.VERSION_CODES.M)
129                 public void onClick(View v) {
130                     requestPermissions(new String[
] {READ_CONTACTS}, REQUEST_READ_CONTACTS);
131                 }
132             });
133     } else {
134         requestPermissions(new String[] {READ_CONTACTS}
, REQUEST_READ_CONTACTS);
135     }
136     return false;
137 }
138
139 /**
140  * Callback received when a permissions request has
been completed.
141  */
142 @Override

```

```
143     public void onRequestPermissionsResult(int requestCode
144     , @NonNull String[] permissions,
145                                           @NonNull int[]
146     grantResults) {
147         if (requestCode == REQUEST_READ_CONTACTS) {
148             if (grantResults.length == 1 && grantResults[0
149 ] == PackageManager.PERMISSION_GRANTED) {
150                 populateAutoComplete();
151             }
152         }
153     }
154     /**
155     * Attempts to sign in or register the account
156     specified by the login form.
157     * If there are form errors (invalid email, missing
158     fields, etc.), the
159     * errors are presented and no actual login attempt is
160     made.
161     */
162     private void attemptLogin() {
163         // Reset errors.
164         mEmailView.setError(null);
165         mPasswordView.setError(null);
166
167         // Store values at the time of the login attempt.
168         final String email = mEmailView.getText().toString
169 ();
170         String password = mPasswordView.getText().toString
171 ();
172
173         boolean cancel = false;
174         View focusView = null;
175
176         // Check for a valid password, if the user entered
177         one.
178         if (!TextUtils.isEmpty(password) && !
179 isPasswordValid(password)) {
180             mPasswordView.setError(getString(R.string.
181 error_invalid_password));
182             focusView = mPasswordView;
183             cancel = true;
184         }
185
186         // Check for a valid email address.
187         if (TextUtils.isEmpty(email)) {
188             mEmailView.setError(getString(R.string.
189 error_field_required));
190             focusView = mEmailView;
191             cancel = true;
192         } else if (!isEmailValid(email)) {
193             mEmailView.setError(getString(R.string.
194 error_invalid_email));
195             focusView = mEmailView;
```

```

185         cancel = true;
186     }
187
188     if (cancel) {
189         // There was an error; don't attempt login and
        focus the first
190         // form field with an error.
191         focusView.requestFocus();
192     } else {
193         // Show a progress spinner, and kick off a
        background task to
194         // perform the user login attempt.
195         showProgress(true);
196         auth.signInWithEmailAndPassword(email,
        password)
197             .addOnCompleteListener(LoginActivity.
        this, new OnCompleteListener<AuthResult>() {
198             @Override
199             public void onComplete(@NonNull
        Task<AuthResult> task) {
200                 // If sign in fails, display a
        message to the user. If sign in succeeds
201                 // the auth state listener
        will be notified and logic to handle the
202                 // signed in user can be
        handled in the listener.
203                 showProgress(false);
204                 if (!task.isSuccessful()) {
205                     // there was an error
206                     if (mPasswordView.length()
        < 6) {
207                         mPasswordView.setError
        ("Minimum 6 digit");
208                     } else {
209                         Toast.makeText(
        LoginActivity.this, "Authentication failed, check your
        email and password or sign up", Toast.LENGTH_LONG).show();
210                     }
211                 } else {
212                     Intent intent = new Intent
        (LoginActivity.this, hal_home.class);
213                     Toast.makeText(
        getApplicationContext(), "You are logged in", Toast.
        LENGTH_SHORT).show();
214                     startActivity(intent);
215                     finish();
216                 }
217             }
218         });
219     }
220 }
221
222 private boolean isEmailValid(String email) {
223     //TODO: Replace this with your own logic
224     return email.contains("@");

```

```
225     }
226
227     private boolean isPasswordValid(String password) {
228         //TODO: Replace this with your own logic
229         return password.length() > 4;
230     }
231
232     /**
233      * Shows the progress UI and hides the login form.
234      */
235     @TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
236     private void showProgress(final boolean show) {
237         // On Honeycomb MR2 we have the
238         ViewPropertyAnimator APIs, which allow
239         // for very easy animations. If available, use
240         these APIs to fade-in
241         // the progress spinner.
242         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.
243 HONEYCOMB_MR2) {
244             int shortAnimTime = getResources().getInteger(
245 android.R.integer.config_shortAnimTime);
246
247             mLoginFormView.setVisibility(show ? View.GONE
248 : View.VISIBLE);
249             mLoginFormView.animate().setDuration(
250 shortAnimTime).alpha(
251 show ? 0 : 1).setListener(new
252 AnimatorListenerAdapter() {
253             @Override
254             public void onAnimationEnd(Animator
255 animation) {
256                 mLoginFormView.setVisibility(show ?
257 View.GONE : View.VISIBLE);
258             }
259         });
260         } else {
261             // The ViewPropertyAnimator APIs are not
262             available, so simply show
263             // and hide the relevant UI components.
264             mProgressView.setVisibility(show ? View.
265 VISIBLE : View.GONE);
```

```
264         mLoginFormView.setVisibility(show ? View.GONE
      : View.VISIBLE);
265     }
266 }
267
268 @Override
269 public Loader<Cursor> onCreateLoader(int i, Bundle
bundle) {
270     return new CursorLoader(this,
271         // Retrieve data rows for the device user's
      s 'profile' contact.
272         Uri.withAppendedPath(ContactsContract.
Profile.CONTENT_URI,
273         ContactsContract.Contacts.Data.
CONTENT_DIRECTORY), ProfileQuery.PROJECTION,
274
275         // Select only email addresses.
276         ContactsContract.Contacts.Data.MIMETYPE +
277         " = ?", new String[]{
ContactsContract.CommonDataKinds.Email
278         .CONTENT_ITEM_TYPE},
279
280         // Show primary email addresses first.
      Note that there won't be
281         // a primary email address if the user
      hasn't specified one.
282         ContactsContract.Contacts.Data.IS_PRIMARY
+ " DESC");
283     }
284
285 @Override
286 public void onLoadFinished(Loader<Cursor> cursorLoader
, Cursor cursor) {
287     List<String> emails = new ArrayList<>();
288     cursor.moveToFirst();
289     while (!cursor.isAfterLast()) {
290         emails.add(cursor.getString(ProfileQuery.
ADDRESS));
291         cursor.moveToNext();
292     }
293
294     addEmailsToAutoComplete(emails);
295 }
296
297 @Override
298 public void onLoaderReset(Loader<Cursor> cursorLoader)
{
299
300 }
301
302 private void addEmailsToAutoComplete(List<String>
emailAddressCollection) {
303     //Create adapter to tell the AutoCompleteTextView
      what to show in its dropdown list.
304     ArrayAdapter<String> adapter =
```

```
305         new ArrayAdapter<>(LoginActivity.this,
306             android.R.layout.
            simple_dropdown_item_1line, emailAddressCollection);
307
308         mView.setAdapter(adapter);
309     }
310
311
312     private interface ProfileQuery {
313         String[] PROJECTION = {
314             ContactsContract.CommonDataKinds.Email.
ADDRESS,
315             ContactsContract.CommonDataKinds.Email.
IS_PRIMARY,
316         };
317
318         int ADDRESS = 0;
319         int IS_PRIMARY = 1;
320     }
321 }
322
323
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 /**
4  * Created by Zulfa_K on 20-06-2017.
5  */
6
7 public class DataPerjalanan {
8     public static String origin,destination;
9     public static double originLat,originLong,
10    destinationLat,destinationLong;
11 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.animation.Animator;
4 import android.animation.AnimatorListenerAdapter;
5 import android.annotation.TargetApi;
6 import android.content.pm.PackageManager;
7 import android.support.annotation.NonNull;
8 import android.support.design.widget.Snackbar;
9 import android.support.v7.app.AppCompatActivity;
10 import android.app.LoaderManager.LoaderCallbacks;
11
12 import android.content.CursorLoader;
13 import android.content.Loader;
14 import android.database.Cursor;
15 import android.net.Uri;
16 import android.os.AsyncTask;
17
18 import android.os.Build;
19 import android.os.Bundle;
20 import android.provider.ContactsContract;
21 import android.text.TextUtils;
22 import android.view.KeyEvent;
23 import android.view.View;
24 import android.view.View.OnClickListener;
25 import android.view.inputmethod.EditorInfo;
26 import android.widget.ArrayAdapter;
27 import android.widget.AutoCompleteTextView;
28 import android.widget.Button;
29 import android.widget.EditText;
30 import android.widget.TextView;
31 import android.widget.Toast;
32
33 import com.google.android.gms.tasks.OnCompleteListener;
34 import com.google.android.gms.tasks.Task;
35 import com.google.firebase.auth.AuthResult;
36 import com.google.firebase.auth.FirebaseAuth;
37
38 import java.util.ArrayList;
39 import java.util.List;
40
41 import static android.Manifest.permission.READ_CONTACTS;
42
43 /**
44  * A login screen that offers login via email/password.
45  */
46 public class SignupActivity extends AppCompatActivity
47     implements LoaderCallbacks<Cursor> {
48     /**
49      * Id to identity READ_CONTACTS permission request.
50      */
51     private static final int REQUEST_READ_CONTACTS = 0;
52
53     /**
54      * A dummy authentication store containing known user
```



```
54 names and passwords.
55     * TODO: remove after connecting to a real
    authentication system.
56     */
57     private static final String[] DUMMY_CREDENTIALS = new
String[]{
58         "foo@example.com:hello", "bar@example.com:
world"
59     };
60     /**
61     * Keep track of the login task to ensure we can
cancel it if requested.
62     */
63
64     // UI references.
65     private autoCompleteTextView mEmailView;
66     private EditText mPasswordView;
67     private View mProgressView;
68     private View mLoginFormView;
69
70     private FirebaseAuth auth;
71
72     @Override
73     protected void onCreate(Bundle savedInstanceState) {
74         super.onCreate(savedInstanceState);
75         setContentView(R.layout.activity_signup);
76         // Set up the login form.
77         mEmailView = (AutoCompleteTextView) findViewById(R
.id.email);
78         populateAutoComplete();
79
80         mPasswordView = (EditText) findViewById(R.id.
password);
81
82         Button mEmailSignInButton = (Button) findViewById(
R.id.email_sign_in_button);
83         mEmailSignInButton.setOnClickListener(new
OnClickListener() {
84             @Override
85             public void onClick(View view) {
86                 attemptLogin();
87             }
88         });
89
90         mLoginFormView = findViewById(R.id.login_form);
91         mProgressView = findViewById(R.id.login_progress);
92
93         auth = FirebaseAuth.getInstance();
94     }
95
96     private void populateAutoComplete() {
97         if (!mayRequestContacts()) {
98             return;
99         }
100
```

```
101     getLoaderManager().initLoader(0, null, this);
102 }
103
104     private boolean mayRequestContacts() {
105         if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M)
106         {
107             return true;
108         }
109         if (checkSelfPermission(READ_CONTACTS) ==
110 PackageManager.PERMISSION_GRANTED) {
111             return true;
112         }
113         if (shouldShowRequestPermissionRationale(
114 READ_CONTACTS)) {
115             Snackbar.make(mEmailView, R.string.
116 permission_rationale, Snackbar.LENGTH_INDEFINITE)
117                 .setAction(android.R.string.ok, new
118 View.OnClickListener() {
119                 @Override
120                 @TargetApi(Build.VERSION_CODES.M)
121                 public void onClick(View v) {
122                     requestPermissions(new String[
123 ]{READ_CONTACTS}, REQUEST_READ_CONTACTS);
124                 }
125             });
126         } else {
127             requestPermissions(new String[]{READ_CONTACTS}
128 , REQUEST_READ_CONTACTS);
129         }
130         return false;
131     }
132
133     /**
134      * Callback received when a permissions request has
135      * been completed.
136      */
137     @Override
138     public void onRequestPermissionsResult(int requestCode
139 , @NonNull String[] permissions,
140                                     @NonNull int[]
141 grantResults) {
142         if (requestCode == REQUEST_READ_CONTACTS) {
143             if (grantResults.length == 1 && grantResults[0
144 ] == PackageManager.PERMISSION_GRANTED) {
145                 populateAutoComplete();
146             }
147         }
148     }
149
150     /**
151      * Attempts to sign in or register the account
152      * specified by the login form.
153      * If there are form errors (invalid email, missing
154      * fields, etc.), the
```

```
143     * errors are presented and no actual login attempt is
      made.
144     */
145     private void attemptLogin() {
146         // Reset errors.
147         mEmailView.setError(null);
148         mPasswordView.setError(null);
149
150         // Store values at the time of the login attempt.
151         String email = mEmailView.getText().toString();
152         String password = mPasswordView.getText().toString
      ();
153
154         boolean cancel = false;
155         View focusView = null;
156
157         // Check for a valid password, if the user entered
      one.
158         if (!TextUtils.isEmpty(password) && !
      isPasswordValid(password)) {
159             mPasswordView.setError(getString(R.string.
      error_invalid_password));
160             focusView = mPasswordView;
161             cancel = true;
162         }
163
164         // Check for a valid email address.
165         if (TextUtils.isEmpty(email)) {
166             mEmailView.setError(getString(R.string.
      error_field_required));
167             focusView = mEmailView;
168             cancel = true;
169         } else if (!isEmailValid(email)) {
170             mEmailView.setError(getString(R.string.
      error_invalid_email));
171             focusView = mEmailView;
172             cancel = true;
173         }
174
175         if (cancel) {
176             // There was an error; don't attempt login and
      focus the first
177             // form field with an error.
178             focusView.requestFocus();
179         } else {
180             // Show a progress spinner, and kick off a
      background task to
181             // perform the user login attempt.
182             showProgress(true);
183             auth.createUserWithEmailAndPassword(email,
      password)
184                 .addOnCompleteListener(SignupActivity.
      this, new OnCompleteListener<AuthResult>() {
185                 @Override
186                 public void onComplete(@NonNull
```

```

186 Task<AuthResult> task) {
187     Toast.makeText(SignupActivity.
    this, "createUserWithEmail:onComplete:" + task.
    isSuccessful(), Toast.LENGTH_SHORT).show();
188     // If sign in fails, display a
    message to the user. If sign in succeeds
189     // the auth state listener
    will be notified and logic to handle the
190     // signed in user can be
    handled in the listener.
191     if (!task.isSuccessful()) {
192         Toast.makeText(
    SignupActivity.this, "Authentication failed." + task.
    getException(),
193         Toast.LENGTH_SHORT
    ).show();
194     } else {
195         Toast.makeText(
    getApplicationContext(), "Your account was created.",
    Toast.LENGTH_SHORT).show();
196         finish();
197     }
198     showProgress(false);
199 }
200 });
201 }
202 }
203
204 private boolean isValidEmail(String email) {
205     //TODO: Replace this with your own logic
206     return email.contains("@");
207 }
208
209 private boolean isValidPassword(String password) {
210     //TODO: Replace this with your own logic
211     return password.length() > 4;
212 }
213
214 /**
215  * Shows the progress UI and hides the login form.
216  */
217 @TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
218 private void showProgress(final boolean show) {
219     // On Honeycomb MR2 we have the
    ViewPropertyAnimator APIs, which allow
220     // for very easy animations. If available, use
    these APIs to fade-in
221     // the progress spinner.
222     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.
    HONEYCOMB_MR2) {
223         int shortAnimTime = getResources().getInteger(
    android.R.integer.config_shortAnimTime);
224
225         mLoginFormView.setVisibility(show ? View.GONE
    : View.VISIBLE);

```

```
226         mLoginFormView.animate().setDuration(
shortAnimTime).alpha(
227             show ? 0 : 1).setListener(new
AnimatorListenerAdapter() {
228             @Override
229             public void onAnimationEnd(Animator
animation) {
230                 mLoginFormView.setVisibility(show ?
View.GONE : View.VISIBLE);
231             }
232         });
233
234         mProgressView.setVisibility(show ? View.
VISIBLE : View.GONE);
235         mProgressView.animate().setDuration(
shortAnimTime).alpha(
236             show ? 1 : 0).setListener(new
AnimatorListenerAdapter() {
237             @Override
238             public void onAnimationEnd(Animator
animation) {
239                 mProgressView.setVisibility(show ?
View.VISIBLE : View.GONE);
240             }
241         });
242     } else {
243         // The ViewPropertyAnimator APIs are not
available, so simply show
244         // and hide the relevant UI components.
245         mProgressView.setVisibility(show ? View.
VISIBLE : View.GONE);
246         mLoginFormView.setVisibility(show ? View.GONE
: View.VISIBLE);
247     }
248 }
249
250 @Override
251 public Loader<Cursor> onCreateLoader(int i, Bundle
bundle) {
252     return new CursorLoader(this,
253         // Retrieve data rows for the device user's
'profile' contact.
254         Uri.withAppendedPath(ContactsContract.
Profile.CONTENT_URI,
255             ContactsContract.Contacts.Data.
CONTENT_DIRECTORY), ProfileQuery.PROJECTION,
256
257         // Select only email addresses.
258         ContactsContract.Contacts.Data.MIMETYPE +
259             " = ?", new String[]{
ContactsContract.CommonDataKinds.Email
260             .CONTENT_ITEM_TYPE},
261
262         // Show primary email addresses first.
Note that there won't be
```

```
263         // a primary email address if the user  
264         hasn't specified one.  
264         ContactsContract.Contacts.Data.IS_PRIMARY  
265     + " DESC");  
265     }  
266  
267     @Override  
268     public void onLoadFinished(Loader<Cursor> cursorLoader  
269     , Cursor cursor) {  
269         List<String> emails = new ArrayList<>();  
270         cursor.moveToFirst();  
271         while (!cursor.isAfterLast()) {  
272             emails.add(cursor.getString(ProfileQuery.  
273             ADDRESS));  
273             cursor.moveToNext();  
274         }  
275  
276         addEmailsToAutoComplete(emails);  
277     }  
278  
279     @Override  
280     public void onLoaderReset(Loader<Cursor> cursorLoader)  
281     {  
282     }  
283  
284     private void addEmailsToAutoComplete(List<String>  
285     emailAddressCollection) {  
285         //Create adapter to tell the AutoCompleteTextView  
286         what to show in its dropdown list.  
286         ArrayAdapter<String> adapter =  
287             new ArrayAdapter<>(SignupActivity.this,  
288                 android.R.layout.  
289                 simple_dropdown_item_1line, emailAddressCollection);  
290         mEmailView.setAdapter(adapter);  
291     }  
292  
293  
294     private interface ProfileQuery {  
295         String[] PROJECTION = {  
296             ContactsContract.CommonDataKinds.Email.  
297             ADDRESS,  
298             ContactsContract.CommonDataKinds.Email.  
299             IS_PRIMARY,  
300             };  
300         int ADDRESS = 0;  
301         int IS_PRIMARY = 1;  
302     }  
303 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.util.Log;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.ProgressBar;
10 import android.widget.TextView;
11 import android.widget.Toast;
12
13 import com.google.firebase.database.DataSnapshot;
14 import com.google.firebase.database.DatabaseError;
15 import com.google.firebase.database.DatabaseReference;
16 import com.google.firebase.database.FirebaseDatabase;
17 import com.google.firebase.database.ValueEventListener;
18
19 import java.util.HashMap;
20 import java.util.Map;
21
22 import aplikasiku.navwithdirectionlib.FDBMobel.User;
23
24 public class HelloFirebaseDB extends AppCompatActivity {
25     FirebaseDatabase mFirebaseInstance;
26     DatabaseReference tUser,tLokasi;
27     ValueEventListener tUserListener;
28
29
30     Button butInsert, butUpdate;
31     EditText edName,edStatus,edNameToUpdate,
32     edNameValueUpdate;
33     ProgressBar prgInsert;
34     TextView tvReadResult;
35
36     String readresult;
37     String keyUpdate;
38     User userDataUpdate;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_hello_firebase_db)
44     };
45
46     init();
47     initListener();
48
49     private void initListener() {
50         butInsert.setOnClickListener(new View.
51         OnClickListener() {
52             @Override
53             public void onClick(View v) {
54                 insertData();
55             }
56         });
57     }
58 }
```

```

53     }
54     });
55
56     //selectData();
57     tUserListener = new ValueEventListener() {
58         @Override
59         public void onDataChange(DataSnapshot
dataSnapshot) {
60             readresult="";
61             for(DataSnapshot user : dataSnapshot.
getChildren()){
62                 User usr = user.getValue(User.class);
63                 readresult += user.getKey();
64                 //readresult += " "+ usr.name +", "+
usr.status +". ";
65                 tvReadResult.setText(readresult);
66             }
67         }
68
69         @Override
70         public void onCancelled(DatabaseError
databaseError) {
71             }
72     };
73
74     tUser.addValueEventListener(tUserListener);
75
76     butUpdate.setOnClickListener(new View.
OnClickListener() {
77         @Override
78         public void onClick(View v) {
79             if(!edNameToUpdate.getText().toString().
equals("")){
80                 updateData(edNameToUpdate.getText().
toString(), edNameValueUpdate.getText().toString());
81             }
82         }
83     });
84
85 }
86
87 private void updateData(String valueToUpdate, String
valueUpdate) {
88     prgInsert.setVisibility(View.VISIBLE);
89     mFirebaseInstance.getReference("users").
addListenerForSingleValueEvent(new ValueEventListener() {
90         @Override
91         public void onDataChange(DataSnapshot
dataSnapshot) {
92             for(DataSnapshot data : dataSnapshot.
getChildren()){
93                 User usr = data.getValue(User.class);
94                 if(usr.name.equals(edNameToUpdate.
getText().toString())){
95                     keyUpdate = data.getKey();

```



```

96         userDataUpdate = usr;
97
98         Map<String, Object> childUpdates =
    new HashMap<>();
99         childUpdates.put("/") + keyUpdate,
    new User(edNameValueUpdate.getText().toString(),
    userDataUpdate.status).toMap());
100
101         tUser.updateChildren(childUpdates,
    new DatabaseReference.CompletionListener() {
102             @Override
103             public void onComplete(
    DatabaseError databaseError, DatabaseReference
    databaseReference) {
104                 prgInsert.setVisibility(
    View.INVISIBLE);
105                 if(databaseError==null){
106                     Toast.makeText(
    getApplicationContext(), "update ok", Toast.LENGTH_SHORT).
    show();
107                 }
108                 else{
109                     Toast.makeText(
    getApplicationContext(), "update fail", Toast.LENGTH_SHORT
    ).show();
110                 }
111             }
112         });
113         break;
114     }
115 }
116 }
117
118     @Override
119     public void onCancelled(DatabaseError
    databaseError) {
120
121     }
122 });
123 }
124
125     private void insertData() {
126         String name = edName.getText().toString();
127         String status = edStatus.getText().toString();
128         if(!name.equals("") && !status.equals("")){
129             prgInsert.setVisibility(View.VISIBLE);
130             String uid = tUser.push().getKey();
131             tUser.child(uid).setValue(new User(name,
    status), new DatabaseReference.CompletionListener() {
132                 @Override
133                 public void onComplete(DatabaseError
    databaseError, DatabaseReference databaseReference) {
134                     prgInsert.setVisibility(View.INVISIBLE
    );
135                     if(databaseError==null){

```

```
136         Toast.makeText(
            getApplicationContext(), "insert ok", Toast.LENGTH_SHORT).
            show();
137     }
138     else{
139         Toast.makeText(
            getApplicationContext(), "insert fail", Toast.LENGTH_SHORT
            ).show();
140     }
141 }
142 });
143 }
144 }
145
146     private void init() {
147         butInsert = (Button) findViewById(R.id.
            buttonInsert);
148         edName = (EditText) findViewById(R.id.
            editTextDBname);
149         edStatus = (EditText) findViewById(R.id.
            editTextDBstts);
150         prgInsert = (ProgressBar) findViewById(R.id.
            create_progress);
151         tvReadResult = (TextView) findViewById(R.id.
            textView2);
152         butUpdate = (Button) findViewById(R.id.
            buttonUpdate);
153         edNameToUpdate = (EditText) findViewById(R.id.
            editTextNameToUpdate);
154         edNameValueUpdate = (EditText) findViewById(R.id.
            editTextValueUpdate);
155
156         mFirebaseInstance = FirebaseDatabase.getInstance()
            ;
157         tUser = mFirebaseInstance.getReference("users");
158         tLokasi = mFirebaseInstance.getReference("
            locations");
159     }
160 }
161
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.app.ProgressDialog;
4 import android.content.ContentResolver;
5 import android.content.Intent;
6 import android.graphics.Bitmap;
7 import android.net.Uri;
8 import android.provider.MediaStore;
9 import android.support.annotation.NonNull;
10 import android.support.v7.app.AppCompatActivity;
11 import android.os.Bundle;
12 import android.view.View;
13 import android.webkit.MimeTypeMap;
14 import android.widget.Button;
15 import android.widget.EditText;
16 import android.widget.ImageView;
17 import android.widget.Toast;
18
19 import com.google.android.gms.maps.model.LatLng;
20 import com.google.android.gms.tasks.OnFailureListener;
21 import com.google.android.gms.tasks.OnSuccessListener;
22 import com.google.firebase.database.DatabaseReference;
23 import com.google.firebase.database.FirebaseDatabase;
24 import com.google.firebase.storage.FirebaseStorage;
25 import com.google.firebase.storage.OnProgressListener;
26 import com.google.firebase.storage.StorageReference;
27 import com.google.firebase.storage.UploadTask;
28
29 import java.io.IOException;
30
31 import aplikasiku.navwithdirectionlib.FDBMobel.Hospital;
32
33 public class InsertRsFirebase extends AppCompatActivity {
34
35     // Folder path for Firebase Storage.
36     String Storage_Path = "All_Image_Uploads/";
37
38     // Root Database Name for Firebase Database.
39     public static final String Database_Path = "hospital";
40
41     // Creating button.
42     Button ChooseButton, UploadButton, DisplayImageButton;
43
44     // Creating EditText.
45     EditText RSname, RSLok, RSLat, RSLong ;
46
47     // Creating ImageView.
48     ImageView SelectImage;
49
50     // Creating URI.
51     Uri FilePathUri;
52
53     // Creating StorageReference and DatabaseReference
54     object.
55     StorageReference storageReference;
```

```
55 DatabaseReference databaseReference;
56
57 // Image request code for onActivityResult() .
58 int Image_Request_Code = 7;
59
60 ProgressDialog progressDialog ;
61
62 @Override
63 protected void onCreate(Bundle savedInstanceState) {
64     super.onCreate(savedInstanceState);
65     setContentView(R.layout.
activity_insert_rs_firebase);
66
67     // Assign FirebaseStorage instance to
storageReference.
68     storageReference = FirebaseStorage.getInstance().
getReference();
69
70     // Assign FirebaseDatabase instance with root
database name.
71     databaseReference = FirebaseDatabase.getInstance()
.getReference(Database_Path);
72
73     //Assign ID'S to button.
74     ChooseButton = (Button)findViewById(R.id.
ButtonChooseImage);
75     UploadButton = (Button)findViewById(R.id.
ButtonUploadImage);
76
77
78     // Assign ID's to EditText.
79     RSname = (EditText)findViewById(R.id.
editTextRSname);
80     RSLok = (EditText)findViewById(R.id.editTextRSlloc)
;
81     RSLat = (EditText)findViewById(R.id.editTextRSlat)
;
82     RSLong = (EditText)findViewById(R.id.
editTextRSlong);
83
84     // Assign ID'S to image view.
85     SelectImage = (ImageView)findViewById(R.id.
ShowImageView);
86
87     // Assigning Id to ProgressDialog.
88     progressDialog = new ProgressDialog(
InsertRsFirebase.this);
89
90     // Adding click listener to Choose image button.
91     ChooseButton.setOnClickListener(new View.
OnClickListener() {
92         @Override
93         public void onClick(View view) {
94
95             // Creating intent.
```

```
96         Intent intent = new Intent();
97
98         // Setting intent type as image to select
image from phone storage.
99         intent.setType("image/*");
100        intent.setAction(Intent.ACTION_GET_CONTENT
);
101        startActivityForResult(Intent.
createChooser(intent, "Please Select Image"),
Image_Request_Code);
102
103    }
104    });
105
106
107    // Adding click listener to Upload image button.
108    UploadButton.setOnClickListener(new View.
OnClickListener() {
109        @Override
110        public void onClick(View view) {
111
112            // Calling method to upload selected image
on Firebase storage.
113            UploadImageFileToFirebaseStorage();
114
115        }
116    });
117
118    }
119
120    @Override
121    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
122
123        super.onActivityResult(requestCode, resultCode,
data);
124
125        if (requestCode == Image_Request_Code &&
resultCode == RESULT_OK && data != null && data.getData()
!= null) {
126
127            FilePathUri = data.getData();
128
129            try {
130
131                // Getting selected image into Bitmap.
132                Bitmap bitmap = MediaStore.Images.Media.
getBitmap(getContentResolver(), FilePathUri);
133
134                // Setting up bitmap selected image into
ImageView.
135                SelectImage.setImageBitmap(bitmap);
136
137                // After selecting image change choose
button above text.
```

```
138         ChooseButton.setText("Image Selected");
139     }
140     }
141     catch (IOException e) {
142     }
143     e.printStackTrace();
144     }
145     }
146 }
147
148 // Creating Method to get the selected image file
149 // Extension from File Path URI.
150 public String GetFileExtension(Uri uri) {
151     ContentResolver contentResolver =
152     getContentResolver();
153     MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton
154     ();
155     // Returning the file Extension.
156     return mimeTypeMap.getExtensionFromMimeType(
157     contentResolver.getType(uri)) ;
158 }
159
160 // Creating UploadImageFileToFirebaseStorage method to
161 // upload image on storage.
162 public void UploadImageFileToFirebaseStorage() {
163     // Checking whether FilePathUri Is empty or not.
164     if (FilePathUri != null) {
165     }
166     // Setting progressDialog Title.
167     progressDialog.setTitle("Image is Uploading
168     ...");
169     // Showing progressDialog.
170     progressDialog.show();
171     // Creating second StorageReference.
172     StorageReference storageReference2nd =
173     storageReference.child(Storage_Path + System.
174     currentTimeMillis() + "." + GetFileExtension(FilePathUri))
175     ;
176     // Adding addOnSuccessListener to second
177     // StorageReference.
178     storageReference2nd.putFile(FilePathUri)
179     .addOnSuccessListener(new
180     OnSuccessListener<UploadTask.TaskSnapshot>() {
181         @Override
182         public void onSuccess(UploadTask.
183         TaskSnapshot taskSnapshot) {
184         }
185     })
186 }
```

```

181 // Hiding the progressDialog
    after done uploading.
182 progressDialog.dismiss();
183
184 // Showing toast message after
    done uploading.
185
186
187 String namaRS = RSname.getText
    ().toString().trim();
188 String alamat = RSLok.getText(
    ).toString().trim();
189
190 double ordinat_lat = Double.
    parseDouble(RSLat.getText().toString().trim());
191 double ordinat_long = Double.
    parseDouble(RSLong.getText().toString().trim());
192 LatLng ordinat_alamat = new
    LatLng(-7.686646, 110.341827);
193
194 @SuppressWarnings("
    VisibleForTests")
195 Hospital imageUploadInfo = new
    Hospital(namaRS, taskSnapshot.getDownloadUrl().toString(
    ),alamat, ordinat_lat, ordinat_long);
196
197
198 // Getting image upload ID.
199 String ImageUploadId =
    databaseReference.push().getKey();
200
201 // Adding image upload id s
    child element into databaseReference.
202 databaseReference.child(
    ImageUploadId).setValue(imageUploadInfo);
203
204 Toast.makeText(
    getApplicationContext(), "Data RS Uploaded Successfully ",
    Toast.LENGTH_LONG).show();
205     }
206     })
207 // If something goes wrong .
208 .addOnFailureListener(new
    OnFailureListener() {
209     @Override
210     public void onFailure(@NonNull
    Exception exception) {
211
212 // Hiding the progressDialog.
213 progressDialog.dismiss();
214
215 // Showing exception erro
    message.
216 Toast.makeText(
    InsertRsFirebase.this, exception.getMessage(), Toast.

```

```
216 LENGTH_LONG).show();
217         }
218     })
219
220     // On progress change upload time.
221     .addOnProgressListener(new
222     OnProgressListener<UploadTask.TaskSnapshot>() {
223         @Override
224         public void onProgress(UploadTask.
225         TaskSnapshot taskSnapshot) {
226             // Setting progressDialog
227             Title.
228             progressDialog.setTitle("Image
229             is Uploading...");
230         }
231     }
232     else {
233         Toast.makeText(InsertRsFirebase.this, "Please
234         Select Image or Add Image Name", Toast.LENGTH_LONG).show()
235         ;
236     }
237 }
238
```



```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.support.v7.widget.RecyclerView;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.TextView;
8
9 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
10
11 /**
12  * Created by Zulfa_K on 30-05-2017.
13  */
14
15 public class AdminListTLAdapter extends RecyclerView.
Adapter<AdminListTLAdapter.ViewHolder> {
16     private NewSkemaTL[] mDataset;
17
18     public static class ViewHolder extends RecyclerView.
ViewHolder{
19         public TextView mTextViewTLName,mTextViewTLAlamat,
mTextViewTLProv,mTextViewTLLat,mTextViewTLLong;
20
21         public ViewHolder(View v){
22             super(v);
23             mTextViewTLName = (TextView) v.findViewById(R.
id.textViewListTLName);
24             mTextViewTLAlamat = (TextView) v.findViewById(R
.id.textViewListTLAlamat);
25             mTextViewTLLat = (TextView) v.findViewById(R.id
.textViewListTLLat);
26             mTextViewTLLong = (TextView) v.findViewById(R.
id.textViewListTLLong);
27         }
28     }
29
30     public AdminListTLAdapter(NewSkemaTL[] myDataset) {
31         mDataset = myDataset;
32     }
33
34     //implement method
35     @Override
36     public AdminListTLAdapter.ViewHolder onCreateViewHolder
(ViewGroup parent, int viewType) {
37         View rowView = LayoutInflater.from(parent.
getContext()).inflate(R.layout.item_list_lampu_admin,parent
,false);
38         ViewHolder vh = new ViewHolder(rowView);
39         return vh;
40     }
41
42     @Override
43     public void onBindViewHolder(ViewHolder holder, int
position) {
44         holder.mTextViewTLName.setText(mDataset[position].
```

```
44 TLname);
45     holder.mTextViewTLAlamat.setText(mDataset[position]
    .TLloc);
46     holder.mTextViewTLLat.setText(Double.toString(
    mDataset[position].TLlat));
47     holder.mTextViewTLLong.setText(Double.toString(
    mDataset[position].TLlong));
48     }
49
50     @Override
51     public int getItemCount() {
52         return mDataset.length;
53     }
54 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.location.Address;
6 import android.location.Geocoder;
7 import android.location.Location;
8 import android.location.LocationListener;
9 import android.location.LocationManager;
10 import android.support.v7.app.AppCompatActivity;
11 import android.os.Bundle;
12 import android.text.TextUtils;
13 import android.view.View;
14 import android.widget.AdapterView;
15 import android.widget.Button;
16 import android.widget.CheckBox;
17 import android.widget.EditText;
18 import android.widget.Spinner;
19 import android.widget.Toast;
20
21 import com.google.firebase.auth.FirebaseAuth;
22 import com.google.firebase.database.DataSnapshot;
23 import com.google.firebase.database.DatabaseError;
24 import com.google.firebase.database.DatabaseReference;
25 import com.google.firebase.database.FirebaseDatabase;
26 import com.google.firebase.database.ValueEventListener;
27
28 import java.util.List;
29 import java.util.Locale;
30
31 import aplikasiku.navwithdirectionlib.FDBMobel.Area;
32 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
33 import aplikasiku.navwithdirectionlib.FDBMobel.
    NewSkemaTLpoint;
34 import aplikasiku.navwithdirectionlib.FDBMobel.
    NewSkemaTLpos;
35
36 public class SkemaTLbaruActivity extends AppCompatActivity
    {
37
38     FirebaseDatabase mFirebaseInstance;
39     DatabaseReference mDbTL;
40
41     Spinner spinnerTLarea;
42     EditText edTLname, edTLloc, edTLlat, edTLlong;
43     CheckBox cbTLb, cbTLt, cbTLu, cbTLs;
44     Button butInsertTL, butGetLocTL;
45
46     String uid_area;
47     GPS gps;
48
49     @Override
50     protected void onCreate(Bundle savedInstanceState) {
51         super.onCreate(savedInstanceState);
52         setContentView(R.layout.activity_skema_tlbaru);
```

```

53
54     mFirebaseInstance = FirebaseDatabase.getInstance()
;
55     mDbTL = mFirebaseInstance.getReference("new-tl");
56
57     init();
58 }
59
60     private void init() {
61         spinnerTLarea = (Spinner) findViewById(R.id.
spinnerTLarea);
62         edTLname = (EditText) findViewById(R.id.
editTextTLname);
63         edTLloc = (EditText) findViewById(R.id.
editTextTLloc);
64         edTLlat = (EditText) findViewById(R.id.
editTextTLlat);
65         edTLlong = (EditText) findViewById(R.id.
editTextTLlong);
66         cbTLb = (CheckBox) findViewById(R.id.checkBoxTLb);
67         cbTLt = (CheckBox) findViewById(R.id.checkBoxTLt);
68         cbTLu = (CheckBox) findViewById(R.id.checkBoxTLu);
69         cbTLs = (CheckBox) findViewById(R.id.checkBoxTLs);
70         butInsertTL = (Button) findViewById(R.id.
buttonAddTL);
71         butGetLocTL = (Button) findViewById(R.id.
buttonGetLocTL);
72
73         spinnerTLarea.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
74             @Override
75             public void onItemSelected(AdapterView<?>
parent, View view, int position, long id) {
76                 mFirebaseInstance.getReference("area").
addListenerForSingleValueEvent(new ValueEventListener() {
77                     @Override
78                     public void onDataChange(DataSnapshot
dataSnapshot) {
79                         for(DataSnapshot area :
dataSnapshot.getChildren()){
80                             Area x = area.getValue(Area.
class);
81                             if(x.area_name.equals(
spinnerTLarea.getSelectedItem().toString())){
82                                 uid_area = area.getKey();
83                                 Toast.makeText(
getApplicationContext(), uid_area, Toast.LENGTH_SHORT).
show();
84                                 }
85                             }
86                         }
87
88                     @Override
89                     public void onCancelled(DatabaseError
databaseError) {

```

```

90
91         }
92     });
93 }
94
95     @Override
96     public void onNothingSelected(AdapterView<?>
parent) {
97
98     }
99     });
100
101     butGetLocTL.setOnClickListener(new View.
OnClickListener() {
102         @Override
103         public void onClick(View v) {
104             getGPSLoc();
105         }
106     });
107
108     butInsertTL.setOnClickListener(new View.
OnClickListener() {
109         @Override
110         public void onClick(View v) {
111             if(inputIsComplete()){
112                 String tlname = edTLname.getText().
toString();
113                 String tlloc = edTLloc.getText().
toString();
114                 double tllat = Double.parseDouble(
edTLlat.getText().toString());
115                 double tllong = Double.parseDouble(
edTLlong.getText().toString());
116
117                 int tlcount=0;
118                 if(cbTLb.isChecked()){tlcount++;}
119                 if(cbTLt.isChecked()){tlcount++;}
120                 if(cbTLu.isChecked()){tlcount++;}
121                 if(cbTLs.isChecked()){tlcount++;}
122
123                 insertTL(uid_area,tlname,tlloc,tllat,
tllong,tlcount,
124                     cbTLb.isChecked(),cbTLt.
isChecked(),cbTLu.isChecked(),cbTLs.isChecked());
125             }
126             else{
127                 Toast.makeText(getApplicationContext()
, "Please complete all input column", Toast.LENGTH_SHORT).
show();
128             }
129         }
130     });
131 }
132
133     private boolean inputIsComplete(){

```

```

134     boolean state = true;
135     if(TextUtils.isEmpty(edTLname.getText())){
136         state = false;
137     }
138     else if(TextUtils.isEmpty(edTLloc.getText())){
139         state = false;
140     }
141     else if(TextUtils.isEmpty(edTLlat.getText())){
142         state = false;
143     }
144     else if(TextUtils.isEmpty(edTLlong.getText())){
145         state = false;
146     }
147     else if(!cbTLb.isChecked() && !cbTLt.isChecked()
148     && !cbTLu.isChecked() && !cbTLs.isChecked()){
149         state = false;
150     }
151     if(!TextUtils.isEmpty(edTLlat.getText())){
152         try{Double.parseDouble(edTLlat.getText().
153         toString());}
154         catch(Exception e){
155             state = false;
156             Toast.makeText(getApplicationContext(), "
157             Latitude should be double value", Toast.LENGTH_SHORT).show
158             ();
159         }
160     }
161     if(!TextUtils.isEmpty(edTLlong.getText())){
162         try{Double.parseDouble(edTLlong.getText().
163         toString());}
164         catch(Exception e){
165             state = false;
166             Toast.makeText(getApplicationContext(), "
167             Longitude should be double value", Toast.LENGTH_SHORT).
168             show();
169         }
170     }
171     return state;
172 }
173
174 private void insertTL(String uid_area, String TLname,
175 String TLloc, double TLlat, double TLlong, int TLcount,
176 boolean isTLb, boolean isTLt, boolean isTLu, boolean isTLs
177 ) {
178     final ProgressDialog pdDialog = ProgressDialog.
179     show(SkemaTLbaruActivity.this, "Insert TL", "Please wait
180     ...",false,false);
181
182     NewSkemaTLpoint TLb, TLt, TLu, TLs;
183     if(isTLb){TLb = new NewSkemaTLpoint("barat", false
184     );}
185     else{ TLb = null;}

```

```

176
177         if(isTLt){TLt = new NewSkemaTLpoint("timur", false
178         );}
179         else{ TLt = null;}
180         if(isTLu){TLu = new NewSkemaTLpoint("utara", false
181         );}
182         else{ TLu = null;}
183         if(isTLs){TLs = new NewSkemaTLpoint("selatan",
184         false);}
185         else{ TLs = null;}
186         final String uid_tl = mDbTL.push().getKey();
187         mDbTL.child(uid_area).child(uid_tl).setValue(new
188         NewSkemaTL(TLname, TLloc, TLlat, TLlong, TLcount, false,
189         false, new NewSkemaTLpos(TLb, TLt, TLu, TLs)), new
190         DatabaseReference.CompletionListener() {
191             @Override
192             public void onComplete(DatabaseError
193             databaseError, DatabaseReference databaseReference) {
194                 if(databaseError==null){
195                     pdDialog.dismiss();
196                     clearInput();
197                     Toast.makeText(getApplicationContext()
198                     , "insert tl ok", Toast.LENGTH_SHORT).show();
199                 }
200                 else{
201                     Toast.makeText(getApplicationContext()
202                     , "insert tl fail", Toast.LENGTH_SHORT).show();
203                 }
204             }
205         });
206     }
207
208     private void getGPSLoc() {
209         gps = new SkemaTLbaruActivity.GPS();
210         if (gps.isGpsEnable()) {
211             gps.requestNewLocation();
212         } else {
213             Toast.makeText(getApplicationContext(), "GPS
214             feature not detected / available.", Toast.LENGTH_SHORT).
215             show();
216         }
217     }
218
219     private class GPS {
220         String gpsJalan, gpsDesa, gpsKecamatan,
221         gpsKabupaten, gpsProvinsi, gpsNegara, alamat;
222         Double latitude, longitude;
223         private LocationManager locationManager = null;
224         private LocationListener locationListener = null;
225         ProgressDialog pD1;
226
227         public GPS() {

```

```

219     locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);
220     }
221
222     public void requestNewLocation() {
223         locationManager = new SkemaTLbaruActivity.GPS
    .MyLocationListener();
224         locationManager.requestLocationUpdates(
    LocationManager.GPS_PROVIDER, 5000, 10, locationManager);
225         pD1 = ProgressDialog.show(SkemaTLbaruActivity.
    this, "Mencari Lokasi Anda", "Harap tunggu...", true,
    false);
226     }
227
228     public boolean isGpsEnable(){
229         if(locationManager.isProviderEnabled(
    LocationManager.GPS_PROVIDER)){ return true; }
230         else{ return false; }
231     }
232
233     private class MyLocationListener implements
    LocationListener{
234         @Override
235         public void onLocationChanged(Location
    location) {
236             //get long and lat
237             longitude = location.getLongitude();
238             latitude = location.getLatitude();
239
240             //get city name
241             Geocoder geo = new Geocoder(
    getApplicationContext(), Locale.getDefault());
242             List<Address> address;
243
244             try{
245                 address = geo.getFromLocation(location
    .getLatitude(), location.getLongitude(), 1);
246                 if(address.size()>0){
247                     gpsJalan = address.get(0).
    getAddressLine(0);
248                     gpsDesa = address.get(0).
    getSubLocality();
249                     gpsKecamatan = address.get(0).
    getLocality();
250                     gpsKabupaten = address.get(0).
    getSubAdminArea();
251                     gpsProvinsi = address.get(0).
    getAdminArea();
252                     gpsNegara = address.get(0).
    getCountryName();
253                     alamat = gpsJalan + ", " + gpsDesa
    + ", " + gpsKecamatan + ". ";
254                     //alamat+= gpsKabupaten + ", " +
    gpsProvinsi + ", " + gpsNegara;
255

```



```
256         pD1.dismiss();
257
258         edTLloc.setText(alamat);
259         edTLlat.setText(latitude.toString(
260     ));
261         edTLlong.setText (longitude.
262     toString());
263         locationManager.removeUpdates(
264     locationManagerListener);
265     }
266     }catch(Exception e){
267         String error = e.getMessage();
268         Toast.makeText(getApplicationContext()
269     , error, Toast.LENGTH_SHORT).show();
270     }
271     }
272     @Override
273     public void onProviderEnabled(String provider)
274     {
275     }
276     @Override
277     public void onProviderDisabled(String provider
278     ) {
279     }
280     @Override
281     public void onStatusChanged(String provider,
282     int status, Bundle extras) {
283     }
284     }
285     }
286     private void clearInput(){
287         edTLname.setText("");edTLname.clearFocus();
288         edTLloc.setText("");edTLloc.clearFocus();
289         edTLlat.setText("");edTLlat.clearFocus();
290         edTLlong.setText("");edTLlong.clearFocus();
291     }
292     cbTLb.setChecked(false);
293     cbTLt.setChecked(false);
294     cbTLu.setChecked(false);
295     cbTLs.setChecked(false);
296     }
297     }
298 }
299
300
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.content.Context;
4 import android.support.v7.widget.RecyclerView;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ImageView;
9 import android.widget.TextView;
10
11 import com.bumptech.glide.Glide;
12
13 import java.util.ArrayList;
14 import java.util.ConcurrentModificationException;
15 import java.util.List;
16
17 import aplikasiku.navwithdirectionlib.FDBMobel.Hospital;
18
19 /**
20  * Created by Zulfa_K on 23-04-2017.
21  */
22 public class RSRecyclerViewAdapter extends RecyclerView.
Adapter<RSRecyclerViewAdapter.ViewHolder> {
23     private List<Hospital> mDataset;
24     CustomItemClickListener listener;
25     Context context;
26
27     public static class ViewHolder extends RecyclerView.
ViewHolder{
28         public TextView mTextViewRsName;
29         public TextView mTextViewRsLoc;
30         public ImageView mImageViewRs;
31
32         public ViewHolder(View v){
33             super(v);
34             mTextViewRsName = (TextView) v.findViewById(R.
id.textViewRSName);
35             mTextViewRsLoc = (TextView) v.findViewById(R.id
.textViewRSLoc);
36             mImageViewRs = (ImageView) v.findViewById(R.id.
imageViewRSImage);
37         }
38     }
39
40     public RSRecyclerViewAdapter( Context contextRS, List<
Hospital> myDataset, CustomItemClickListener list) {
41         mDataset = myDataset;
42         listener = list;
43         context = contextRS;
44
45     }
46
47     //implement method
48     @Override
49     public RSRecyclerViewAdapter.ViewHolder
```

```

49 onCreateViewHolder(ViewGroup parent, int viewType) {
50     View rowView = LayoutInflater.from(parent.
        getContext()).inflate(R.layout.cusview_listrs,parent,false
        );
51     final ViewHolder vh = new ViewHolder(rowView);
52     rowView.setOnClickListener(new View.
        OnClickListener() {
53         @Override
54         public void onClick(View v) {
55             listener.onItemClick(v, vh.
        getAdapterPosition());
56         }
57     });
58     return vh;
59 }
60
61 @Override
62 public void onBindViewHolder(ViewHolder holder, int
        position) {
63     holder.mTextViewRsName.setText(mDataset.get(
        position).RsName);
64     holder.mTextViewRsLoc.setText(mDataset.get(
        position).RsLoc);
65     Glide.with(context).load(mDataset.get(position).
        RsImageUrl).into(holder.mImageViewRs);
66 //     if(position==0){
67 //         holder.mImageViewRs.setBackgroundResource(R.
        drawable.gbrsardjito);
68 //     }
69 //     else if(position==1){
70 //         holder.mImageViewRs.setBackgroundResource(R.
        drawable.pantirapih);
71 //     }
72 //     else if(position==2){
73 //         holder.mImageViewRs.setBackgroundResource(R.
        drawable.bethesda);
74 //     }
75 }
76
77 @Override
78 public int getItemCount() {
79     return mDataset.size();
80 }
81
82
83 public interface CustomItemClickListener{
84     public void onItemClick(View v, int position);
85 }
86 }
87

```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.app.ProgressDialog;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.support.v7.widget.LinearLayoutManager;
7 import android.support.v7.widget.RecyclerView;
8 import android.view.View;
9 import android.widget.AdapterView;
10 import android.widget.CompoundButton;
11 import android.widget.RadioButton;
12 import android.widget.Spinner;
13 import android.widget.Switch;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 import com.google.firebase.database.DataSnapshot;
18 import com.google.firebase.database.DatabaseError;
19 import com.google.firebase.database.DatabaseReference;
20 import com.google.firebase.database.FirebaseDatabase;
21 import com.google.firebase.database.ValueEventListener;
22
23 import aplikasiku.navwithdirectionlib.FDBMobel.Area;
24 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
25
26 public class KontrolerLampuActivity extends
AppCompatActivity {
27
28     Switch switchB, switchT, switchU, switchS;
29     Switch switchPlaySound;
30     RadioButton radioManual, radioAuto;
31     RadioButton radioBR, radioBY, radioBG;
32     RadioButton radioTR, radioTY, radioTG;
33     RadioButton radioUR, radioUY, radioUG;
34     RadioButton radioSR, radioSY, radioSG;
35
36     FirebaseDatabase database;
37     ProgressDialog pd1;
38
39     KontrolerListTLAdapter.DataTL[] dataTL;
40     KontrolerListTLAdapter adapter;
41     RecyclerView rvListTL;
42
43     Spinner spinnerTLarea;
44
45     String uid_area, uid_tl;
46
47     @Override
48     protected void onCreate(Bundle savedInstanceState) {
49         super.onCreate(savedInstanceState);
50         setContentView(R.layout.activity_kontroler_lampu);
51
52         database = FirebaseDatabase.getInstance();
53
54         switchB = (Switch) findViewById(R.id.
```

```
54 switchLampuBarat);
55     switchT = (Switch) findViewById(R.id.
switchLampuTimur);
56     switchU = (Switch) findViewById(R.id.
switchLampuUtara);
57     switchS = (Switch) findViewById(R.id.
switchLampuSelatan);
58     switchPlaySound = (Switch) findViewById(R.id.
switchTLplaysound);
59
60     radioManual = (RadioButton) findViewById(R.id.
radioKontrolerManual);
61     radioAuto = (RadioButton) findViewById(R.id.
radioKontrolerOtomatis);
62
63     radioBR = (RadioButton) findViewById(R.id.
radioLampuBaratRed);
64     radioBY = (RadioButton) findViewById(R.id.
radioLampuBaratYellow);
65     radioBG = (RadioButton) findViewById(R.id.
radioLampuBaratGreen);
66
67     radioTR = (RadioButton) findViewById(R.id.
radioLampuTimurRed);
68     radioTY = (RadioButton) findViewById(R.id.
radioLampuTimurYellow);
69     radioTG = (RadioButton) findViewById(R.id.
radioLampuTimurGreen);
70
71     radioUR = (RadioButton) findViewById(R.id.
radioLampuUtaraRed);
72     radioUY = (RadioButton) findViewById(R.id.
radioLampuUtaraYellow);
73     radioUG = (RadioButton) findViewById(R.id.
radioLampuUtaraGreen);
74
75     radioSR = (RadioButton) findViewById(R.id.
radioLampuSelatanRed);
76     radioSY = (RadioButton) findViewById(R.id.
radioLampuSelatanYellow);
77     radioSG = (RadioButton) findViewById(R.id.
radioLampuSelatanGreen);
78
79     rvListTL = (RecyclerView) findViewById(R.id.
recyclerViewListLampuKontroler);
80     LinearLayoutManager layoutManager = new
LinearLayoutManager(this);
81     rvListTL.setHasFixedSize(true);
82     rvListTL.setLayoutManager(layoutManager);
83
84     spinnerTLarea = (Spinner) findViewById(R.id.
spinnerKontrolerTLarea);
85     spinnerTLarea.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
86         @Override
```

```

87     public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {
88         setComponentEnable(false);
89         radioManual.setEnabled(false);
90         radioAuto.setEnabled(false);
91
92         database.getReference("area").
addListenerForSingleValueEvent(new ValueEventListener() {
93             @Override
94             public void onDataChange(DataSnapshot
dataSnapshot) {
95                 for(DataSnapshot area :
dataSnapshot.getChildren()){
96                     Area x = area.getValue(Area.
class);
97                     if(x.area_name.equals(
spinnerTLarea.getSelectedItem().toString())){
98                         uid_area = area.getKey();
99                         initGetListTL();
100                        Toast.makeText(
getApplicationContext(), "uid_area " + uid_area, Toast.
LENGTH_SHORT).show();
101                    }
102                }
103            }
104
105            @Override
106            public void onCancelled(DatabaseError
databaseError) {
107            }
108        });
109    }
110
111    @Override
112    public void onNothingSelected(AdapterView<?>
parent) {
113    }
114
115    });
116
117    initListener();
118    initListenerRadioRYG();
119 }
120
121
122     private void initGetListTL() {
123         database.getReference("new-tl").child(uid_area).
addListenerForSingleValueEvent(new ValueEventListener() {
124             @Override
125             public void onDataChange(DataSnapshot
dataSnapshot) {
126                 dataTL = new KontrolerListTLAdapter.DataTL
[(int) dataSnapshot.getChildrenCount()];
127                 int count=0;
128                 for(DataSnapshot data : dataSnapshot.

```

```

128 getChildren()){
129     NewSkemaTL tl = data.getValue(
    NewSkemaTL.class);
130     dataTL[count] = new
    KontrolerListTLAdapter.DataTL();
131     dataTL[count].TLname = tl.TLname;
132     dataTL[count].TLloc = tl.TLloc;
133     dataTL[count].uid_tl = data.getKey();
134     count++;
135     }
136
137     adapter = new KontrolerListTLAdapter(
    dataTL, new KontrolerListTLAdapter.ViewHolderClickListener
    () {
138         @Override
139         public void onViewHolderClick(View v,
    int position) {
140             for(int i=0;i<dataTL.length;i++){
141                 dataTL[i].isSelected = false;
142             }
143             dataTL[position].isSelected = true
    ;
144             adapter.setDataTL(dataTL);
145             rvListTL.setAdapter(adapter);
146             uid_tl = dataTL[position].uid_tl;
147             Toast.makeText(
    getApplicationContext(), "uid_tl " + uid_tl, Toast.
    LENGTH_SHORT).show();
148             dataControlerListener();
149         }
150     });
151     rvListTL.setAdapter(adapter);
152 }
153
154 @Override
155 public void onCancelled(DatabaseError
    databaseError) {
156     }
157 }
158 });
159 }
160
161 private void initListenerRadioRYG() {
162     radioBR.setOnClickListener(new View.
    OnClickListener() {
163         @Override
164         public void onClick(View v) {
165             if(radioBR.isChecked()){
166                 radioBY.setChecked(false);
167                 radioBG.setChecked(false);
168                 chooseLampu("b", "r");
169             }
170         }
171     });
172     radioBY.setOnClickListener(new View.

```

```
172 OnClickListener() {
173     @Override
174     public void onClick(View v) {
175         if(radioBY.isChecked()){
176             radioBR.setChecked(false);
177             radioBG.setChecked(false);
178             chooseLampu("b", "y");
179         }
180     }
181 });
182 radioBG.setOnClickListener(new View.
OnClickListener() {
183     @Override
184     public void onClick(View v) {
185         if(radioBG.isChecked()){
186             radioBY.setChecked(false);
187             radioBR.setChecked(false);
188             chooseLampu("b", "g");
189         }
190     }
191 });
192
193 //=====//
194 radioTR.setOnClickListener(new View.
OnClickListener() {
195     @Override
196     public void onClick(View v) {
197         if(radioTR.isChecked()){
198             radioTY.setChecked(false);
199             radioTG.setChecked(false);
200             chooseLampu("t", "r");
201         }
202     }
203 });
204 radioTY.setOnClickListener(new View.
OnClickListener() {
205     @Override
206     public void onClick(View v) {
207         if(radioTY.isChecked()){
208             radioTR.setChecked(false);
209             radioTG.setChecked(false);
210             chooseLampu("t", "y");
211         }
212     }
213 });
214 radioTG.setOnClickListener(new View.
OnClickListener() {
215     @Override
216     public void onClick(View v) {
217         if(radioTG.isChecked()){
218             radioTY.setChecked(false);
219             radioTR.setChecked(false);
220             chooseLampu("t", "g");
221         }

```



```
222     }
223     });
224     //=====//
225
226     radioUR.setOnCheckedChangeListener(new View.
OnCheckedChangeListener() {
227         @Override
228         public void onClick(View v) {
229             if(radioUR.isChecked()){
230                 radioUY.setChecked(false);
231                 radioUG.setChecked(false);
232                 chooseLampu("u", "r");
233             }
234         }
235     });
236     radioUY.setOnCheckedChangeListener(new View.
OnCheckedChangeListener() {
237         @Override
238         public void onClick(View v) {
239             if(radioUY.isChecked()){
240                 radioUR.setChecked(false);
241                 radioUG.setChecked(false);
242                 chooseLampu("u", "y");
243             }
244         }
245     });
246     radioUG.setOnCheckedChangeListener(new View.
OnCheckedChangeListener() {
247         @Override
248         public void onClick(View v) {
249             if(radioUG.isChecked()){
250                 radioUY.setChecked(false);
251                 radioUR.setChecked(false);
252                 chooseLampu("u", "g");
253             }
254         }
255     });
256     //=====//
257
258     radioSR.setOnCheckedChangeListener(new View.
OnCheckedChangeListener() {
259         @Override
260         public void onClick(View v) {
261             if(radioSR.isChecked()){
262                 radioSY.setChecked(false);
263                 radioSG.setChecked(false);
264                 chooseLampu("s", "r");
265             }
266         }
267     });
268     radioSY.setOnCheckedChangeListener(new View.
OnCheckedChangeListener() {
269         @Override
270         public void onClick(View v) {
271             if(radioSY.isChecked()){
```

```

272         radioSR.setChecked(false);
273         radioSG.setChecked(false);
274         chooseLampu("s","y");
275     }
276 }
277 });
278     radioSG.setOnClickListener(new View.
OnClickListener() {
279         @Override
280         public void onClick(View v) {
281             if(radioSG.isChecked()){
282                 radioSY.setChecked(false);
283                 radioSR.setChecked(false);
284                 chooseLampu("s","g");
285             }
286         }
287     });
288 }
289
290 private void initListener() {
291     radioManual.setOnClickListener(new View.
OnClickListener() {
292         @Override
293         public void onClick(View v) {
294             if(radioManual.isChecked()){
295                 pd1 = ProgressDialog.show(
KontrolerLampuActivity.this, "Mengubah mode", "Harap
tunggu...", false, false);
296                 database.getReference("new-tl").child(
uid_area).child(uid_tl).child("TLstate").setValue(false,
new DatabaseReference.CompletionListener() {
297                     @Override
298                     public void onComplete(
DatabaseError databaseError, DatabaseReference
databaseReference) {
299                         pd1.dismiss();
300                         if(databaseError==null){
301                             radioAuto.setChecked(false
);
302                             dataControlerListener();
303                         }
304                     }
305                 });
306             }
307         }
308     });
309
310     radioAuto.setOnClickListener(new View.
OnClickListener() {
311         @Override
312         public void onClick(View v) {
313             if(radioAuto.isChecked()){
314                 pd1 = ProgressDialog.show(
KontrolerLampuActivity.this, "Mengubah mode", "Harap
tunggu...", false, false);

```

```

315         database.getReference("new-tl").child(
uid_area).child(uid_tl).child("TLstate").setValue(true,
new DatabaseReference.CompletionListener() {
316             @Override
317             public void onComplete(
DatabaseError databaseError, DatabaseReference
databaseReference) {
318                 pd1.dismiss();
319                 if(databaseError==null){
320                     radioManual.setChecked(
false);
321                     setComponentEnable(false);
322                 }
323             }
324         });
325     }
326 }
327 });
328
329     switchB.setOnClickListener(new View.
OnClickListener() {
330         @Override
331         public void onClick(View v) {
332             if(switchB.isChecked()){setLampu("b", true
);}
333             else{setLampu("b", false);}
334         }
335     });
336
337     switchT.setOnClickListener(new View.
OnClickListener() {
338         @Override
339         public void onClick(View v) {
340             if(switchT.isChecked()){setLampu("t", true
);}
341             else{setLampu("t", false);}
342         }
343     });
344
345     switchU.setOnClickListener(new View.
OnClickListener() {
346         @Override
347         public void onClick(View v) {
348             if(switchU.isChecked()){setLampu("u", true
);}
349             else{setLampu("u", false);}
350         }
351     });
352
353     switchS.setOnClickListener(new View.
OnClickListener() {
354         @Override
355         public void onClick(View v) {
356             if(switchS.isChecked()){setLampu("s", true
);}

```

```
357         else{setLampu("s", false);}
358     }
359 });
360
361     switchPlaySound.setOnClickListener(new View.
onClickListener() {
362         @Override
363         public void onClick(View v) {
364             if(switchPlaySound.isChecked()){
365                 setPlaySound(true);
366             }
367             else{
368                 setPlaySound(false);
369             }
370         }
371     });
372 }
373
374     private void setPlaySound(boolean playSound) {
375         pd1 = ProgressDialog.show(KontrolerLampuActivity.
this, "Memperbarui kondisi", "Harap tunggu...",false,false
);
376
377         database.getReference("new-tl").child(uid_area).
child(uid_tl).child("TLplaysound").setValue(playSound, new
DatabaseReference.CompletionListener() {
378             @Override
379             public void onComplete(DatabaseError
databaseError, DatabaseReference databaseReference) {
380                 pd1.dismiss();
381                 if(databaseError==null){
382                     Toast.makeText(getApplicationContext()
, "Set sound Ok", Toast.LENGTH_SHORT).show();
383                 }
384                 else{
385                     Toast.makeText(getApplicationContext()
, "Set sound Fail", Toast.LENGTH_SHORT).show();
386                 }
387             }
388         });
389     }
390
391     private void setComponentEnable(boolean setEnable){
392         switchB.setEnabled(setEnable);
393         switchT.setEnabled(setEnable);
394         switchU.setEnabled(setEnable);
395         switchS.setEnabled(setEnable);
396         switchPlaySound.setEnabled(setEnable);
397
398         radioBR.setEnabled(setEnable);radioBY.setEnabled(
setEnable);radioBG.setEnabled(setEnable);
399         radioTR.setEnabled(setEnable);radioTY.setEnabled(
setEnable);radioTG.setEnabled(setEnable);
400         radioUR.setEnabled(setEnable);radioUY.setEnabled(
setEnable);radioUG.setEnabled(setEnable);
```

```

401     radioSR.setEnabled(setEnable);radioSY.setEnabled(
        setEnable);radioSG.setEnabled(setEnable);
402     }
403
404     public void setLampu(String btus, boolean stat){
405         pd1 = ProgressDialog.show(KontrolerLampuActivity.
this, "Memperbarui kondisi", "Harap tunggu...",false,false
        );
406
407         if(btus.equals("b")){ btus="TLb"; }
408         else if(btus.equals("t")){ btus="TLt"; }
409         else if(btus.equals("u")){ btus="TLu"; }
410         else if(btus.equals("s")){ btus="TLs"; }
411         final String newtl = btus;
412         database.getReference("new-tl").child(uid_area).
child(uid_tl).child("TLpos").child(btus).child("stat").
setValue(stat, new DatabaseReference.CompletionListener()
        {
413             @Override
414             public void onComplete(DatabaseError
databaseError, DatabaseReference databaseReference) {
415                 pd1.dismiss();
416                 if(databaseError==null){
417                     Toast.makeText(getApplicationContext()
, newtl + " Ok", Toast.LENGTH_SHORT).show();
418                 }
419                 else{
420                     Toast.makeText(getApplicationContext()
, newtl + " Fail", Toast.LENGTH_SHORT).show();
421                 }
422             }
423         });
424     }
425
426     public void getLampu(String btus, final Switch
switchLamp){
427         if(btus.equals("b")){ btus="TLb"; }
428         else if(btus.equals("t")){ btus="TLt"; }
429         else if(btus.equals("u")){ btus="TLu"; }
430         else if(btus.equals("s")){ btus="TLs"; }
431
432         database.getReference("new-tl").child(uid_area).
child(uid_tl).child("TLpos").child(btus).child("stat").
addValueEventListener(new ValueEventListener() {
433             @Override
434             public void onDataChange(DataSnapshot
dataSnapshot) {
435                 if(dataSnapshot.exists()) {
436                     if (dataSnapshot.getValue().toString()
.equals("true")) {
437                         switchLamp.setChecked(true);
438                     } else {
439                         switchLamp.setChecked(false);
440                     }
441                 }

```

```

442         else{
443             switchLamp.setEnabled(false);
444         }
445     }
446
447     @Override
448     public void onCancelled(DatabaseError
449 databaseError) {
450         }
451     });
452 }
453
454     public void chooseLampu(String btus, String ryg){
455         pd1 = ProgressDialog.show(KontrolerLampuActivity.
456 this, "Memilih lampu", "Harap tunggu...", false, false);
457
458         if(btus.equals("b")){ btus="TLb"; }
459         else if(btus.equals("t")){ btus="TLt"; }
460         else if(btus.equals("u")){ btus="TLu"; }
461         else if(btus.equals("s")){ btus="TLs"; }
462
463         if(ryg.equals("r")){ ryg="red"; }
464         else if(ryg.equals("y")){ ryg="yellow"; }
465         else if(ryg.equals("g")){ ryg="green"; }
466         final String ulamp = ryg;
467
468         database.getReference("new-tl").child(uid_area).
469         child(uid_tl).child("TLpos").child(btus).child("ulamp").
470         setValue(ryg, new DatabaseReference.CompletionListener() {
471             @Override
472             public void onComplete(DatabaseError
473 databaseError, DatabaseReference databaseReference) {
474                 pd1.dismiss();
475                 if(databaseError==null){
476                     Toast.makeText(getApplicationContext()
477 , ulamp + " Ok", Toast.LENGTH_SHORT).show();
478                 }
479                 else{
480                     Toast.makeText(getApplicationContext()
481 , ulamp + " Fail", Toast.LENGTH_SHORT).show();
482                 }
483             }
484         });
485     }
486
487     public void getChooseLampu(String btus){
488         if(btus.equals("b")){ btus="TLb"; }
489         else if(btus.equals("t")){ btus="TLt"; }
490         else if(btus.equals("u")){ btus="TLu"; }
491         else if(btus.equals("s")){ btus="TLs"; }
492
493         final String btusfin = btus;
494
495         database.getReference("new-tl").child(uid_area).

```

```
489 child(uid_tl).child("TLpos").child(btus).child("uLamp").
    addValueEventListener(new ValueEventListener() {
490         @Override
491         public void onDataChange(DataSnapshot
dataSnapshot) {
492             if(dataSnapshot.exists()){
493                 if(dataSnapshot.getValue().toString().
equals("red")){
494                     if(btusfin.equals("TLb")){
495                         radioBR.setChecked(true);
496                         radioBY.setChecked(false);
497                         radioBG.setChecked(false);
498                     }
499                     else if(btusfin.equals("TLt")){
500                         radioTR.setChecked(true);
501                         radioTY.setChecked(false);
502                         radioTG.setChecked(false);
503                     }
504                     else if(btusfin.equals("TLu")){
505                         radioUR.setChecked(true);
506                         radioUY.setChecked(false);
507                         radioUG.setChecked(false);
508                     }
509                     else if(btusfin.equals("TLs")){
510                         radioSR.setChecked(true);
511                         radioSY.setChecked(false);
512                         radioSG.setChecked(false);
513                     }
514                 }
515                 else if(dataSnapshot.getValue().
toString().equals("yellow")){
516                     if(btusfin.equals("TLb")){
517                         radioBR.setChecked(false);
518                         radioBY.setChecked(true);
519                         radioBG.setChecked(false);
520                     }
521                     else if(btusfin.equals("TLt")){
522                         radioTR.setChecked(false);
523                         radioTY.setChecked(true);
524                         radioTG.setChecked(false);
525                     }
526                     else if(btusfin.equals("TLu")){
527                         radioUR.setChecked(false);
528                         radioUY.setChecked(true);
529                         radioUG.setChecked(false);
530                     }
531                     else if(btusfin.equals("TLs")){
532                         radioSR.setChecked(false);
533                         radioSY.setChecked(true);
534                         radioSG.setChecked(false);
535                     }
536                 }
537                 else if(dataSnapshot.getValue().
toString().equals("green")){
538                     if(btusfin.equals("TLb")){
```

```
539         radioBR.setChecked(false);
540         radioBY.setChecked(false);
541         radioBG.setChecked(true);
542     }
543     else if(btusfin.equals("TLt")){
544         radioTR.setChecked(false);
545         radioTY.setChecked(false);
546         radioTG.setChecked(true);
547     }
548     else if(btusfin.equals("TLu")){
549         radioUR.setChecked(false);
550         radioUY.setChecked(false);
551         radioUG.setChecked(true);
552     }
553     else if(btusfin.equals("TLs")){
554         radioSR.setChecked(false);
555         radioSY.setChecked(false);
556         radioSG.setChecked(true);
557     }
558     }
559 }
560 else{
561     if(btusfin.equals("TLb")){
562         radioBR.setEnabled(false);
563         radioBY.setEnabled(false);
564         radioBG.setEnabled(false);
565     }
566     else if(btusfin.equals("TLt")){
567         radioTR.setEnabled(false);
568         radioTY.setEnabled(false);
569         radioTG.setEnabled(false);
570     }
571     else if(btusfin.equals("TLu")){
572         radioUR.setEnabled(false);
573         radioUY.setEnabled(false);
574         radioUG.setEnabled(false);
575     }
576     else if(btusfin.equals("TLs")){
577         radioSR.setEnabled(false);
578         radioSY.setEnabled(false);
579         radioSG.setEnabled(false);
580     }
581     }
582 }
583
584     @Override
585     public void onCancelled(DatabaseError
586 databaseError) {
587     }
588 });
589 }
590
591     public void dataControlerListener(){
592         radioManual.setEnabled(true);
```



```
593         radioAuto.setEnabled(true);
594         database.getReference("new-tl").child(uid_area).
child(uid_tl).child("TLstate").addValueEventListener(new
ValueEventListener() {
595             @Override
596             public void onDataChange(DataSnapshot
dataSnapshot) {
597                 if(dataSnapshot.getValue().toString().
equals("true")){
598                     radioAuto.setChecked(true);
599                     radioManual.setChecked(false);
600                     setComponentEnable(false);
601                 }
602                 else{
603                     radioAuto.setChecked(false);
604                     radioManual.setChecked(true);
605                     setComponentEnable(true);
606                 }
607             }
608
609             @Override
610             public void onCancelled(DatabaseError
databaseError) {
611                 }
612             });
613     });
614
615     database.getReference("new-tl").child(uid_area).
child(uid_tl).child("TLplaysound").addValueEventListener(
new ValueEventListener() {
616         @Override
617         public void onDataChange(DataSnapshot
dataSnapshot) {
618             if(dataSnapshot.getValue().toString().
equals("true")){
619                 switchPlaySound.setChecked(true);
620             }
621             else{
622                 switchPlaySound.setChecked(false);
623             }
624         }
625
626         @Override
627         public void onCancelled(DatabaseError
databaseError) {
628             }
629         });
630
631
632     getLampu("b", switchB);
633     getLampu("t", switchT);
634     getLampu("u", switchU);
635     getLampu("s", switchS);
636
637     getChooseLampu("b");
```

```
638         getChooseLampu("t");
639         getChooseLampu("u");
640         getChooseLampu("s");
641     }
642 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.support.v7.widget.RecyclerView;
4 import android.text.TextUtils;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ImageView;
9 import android.widget.RadioButton;
10 import android.widget.TextView;
11
12 import com.google.firebase.database.DataSnapshot;
13
14 import aplikasiku.navwithdirectionlib.FDBMobel.NewSkemaTL;
15
16 /**
17  * Created by Zulfa_K on 07-10-2017.
18  */
19
20 public class KontrolerListTLAdapter extends RecyclerView.
Adapter<KontrolerListTLAdapter.TLViewHolder> {
21
22     private ViewHolderClickListener
mOnViewHolderClickListener;
23     private DataTL[] dataTL;
24
25     public static class DataTL{
26         String uid_tl;
27         String TLname;
28         String TLloc;
29         boolean isSelected;
30
31         public DataTL(){
32     }
33
34     public KontrolerListTLAdapter(DataTL[] listTL,
ViewHolderClickListener viewHolderClick){
35         this.dataTL = listTL;
36         mOnViewHolderClickListener = viewHolderClick;
37     }
38
39     public void setDataTL(DataTL[] dtTL){
40         dataTL = dtTL;
41     }
42
43     //viewholder
44     public class TLViewHolder extends RecyclerView.
ViewHolder{
45         //ui component item
46         TextView textViewTLname, textViewTLloc;
47         RadioButton radioButtonTLselected;
48
49         public TLViewHolder(final View itemView) {
50             super(itemView);
51             //findviewbyid
```

```

52         textViewTLname = (TextView) itemView.
findViewById(R.id.textView_kontroler_tlname);
53         textViewTLloc = (TextView) itemView.
findViewById(R.id.textView_kontroler_tlloc);
54         radioButtonTLselected = (RadioButton) itemView
.findViewById(R.id.radio_kontroler_tlselected);
55
56         radioButtonTLselected.setOnClickListener(new
View.OnClickListener() {
57             @Override
58             public void onClick(View v) {
59                 radioButtonTLselected.setChecked(false
);
60                 mOnViewHolderClickListener.
onViewHolderClick(itemView, getAdapterPosition());
61             }
62         });
63     }
64 }
65
66     public KontrolerListTLAdapter.TLViewHolder
onCreateViewHolder(ViewGroup parent, int viewType) {
67         View view = LayoutInflater.from(parent.getContext(
)).inflate(R.layout.item_list_lampu_kontroler,parent,false
);
68         final TLViewHolder viewHolder = new TLViewHolder(
view);
69         view.setOnClickListener(new View.OnClickListener()
{
70             @Override
71             public void onClick(View view) {
72                 mOnViewHolderClickListener.
onViewHolderClick(view, viewHolder.getAdapterPosition());
73             }
74         });
75         return viewHolder;
76     }
77
78     @Override
79     public void onBindViewHolder(KontrolerListTLAdapter.
TLViewHolder holder, int position) {
80         holder.textViewTLname.setText(dataTL[position].
TLname);
81         holder.textViewTLloc.setText(dataTL[position].
TLloc);
82         holder.radioButtonTLselected.setChecked(dataTL[
position].isSelected);
83     }
84
85     @Override
86     public int getItemCount() {
87         return dataTL.length;
88     }
89
90     public interface ViewHolderClickListener{

```

```
91         void onViewHolderClick(View v, int position);  
92     }  
93  
94 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.content.pm.PackageManager;
4 import android.support.v4.app.ActivityCompat;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.support.v7.widget.CardView;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.ImageView;
11 import android.widget.TextView;
12
13 import com.google.android.gms.common.api.Status;
14 import com.google.android.gms.location.places.Place;
15 import com.google.android.gms.location.places.ui.
    PlaceAutocompleteFragment;
16 import com.google.android.gms.location.places.ui.
    PlaceSelectionListener;
17 import com.google.android.gms.maps.CameraUpdateFactory;
18 import com.google.android.gms.maps.GoogleMap;
19 import com.google.android.gms.maps.OnMapReadyCallback;
20 import com.google.android.gms.maps.SupportMapFragment;
21 import com.google.android.gms.maps.model.LatLng;
22 import com.google.android.gms.maps.model.LatLngBounds;
23 import com.loopj.android.http.AsyncHttpClient;
24 import com.loopj.android.http.AsyncHttpResponseHandler;
25
26 import org.json.JSONArray;
27 import org.json.JSONObject;
28
29 import cz.msebera.android.httpclient.Header;
30
31 public class SetDestinationActivity extends
    AppCompatActivity {
32
33     String TAG = ".SetDestinationActivity";
34     private GoogleMap mMap;
35     AsyncHttpClient client;
36     ImageView ivCenter;
37     TextView tvOrigin,tvDestination;
38     boolean isCameraMoveListenerSet = false;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_set_destination);
44
45         PlaceAutocompleteFragment autocompleteFragment = (
    PlaceAutocompleteFragment)
46             getSupportFragmentManager().findFragmentById(R.id.
    place_autocomplete_fragment);
47         autocompleteFragment.setHint("Lokasi tujuan");
48
49         autocompleteFragment.setOnPlaceSelectedListener(new
    PlaceSelectionListener() {
```

```

50         @Override
51         public void onPlaceSelected(Place place) {
52             // TODO: Get info about the selected place
53
54             Log.i(TAG, "Place: " + place.getName());
55             Log.i(TAG, "LatLng: " + place.getLatLng().
56                 toString());
57             mMap.animateCamera(CameraUpdateFactory.
58                 newLatLngZoom(place.getLatLng(), 13));
59             if(!isCameraMoveListenerSet) {
60                 mMap.setOnCameraMoveListener(new
61                 GoogleMap.OnCameraMoveListener() {
62                     @Override
63                     public void onCameraMove() {
64                         if (mMap != null) {
65                             double lat = mMap.
66                             getCameraPosition().target.latitude;
67                             double lng = mMap.
68                             getCameraPosition().target.longitude;
69                             GetLocationBy(new LatLng(
70                             lat, lng));
71                         }
72                     }
73                 });
74                 isCameraMoveListenerSet = true;
75             }
76             ivCenter.setVisibility(View.VISIBLE);
77             GetLocationBy(place.getLatLng());
78         }
79     });
80
81     initMap();
82
83     ivCenter = (ImageView) findViewById(R.id.
84     iv_center_location);
85     String origin_address = getIntent().getStringExtra
86     ("origin_address");
87     tvOrigin = (TextView) findViewById(R.id.
88     tv_origin_address);
89     tvOrigin.setText(origin_address);
90     tvDestination = (TextView) findViewById(R.id.
91     tv_des_address);
92 }
93
94 private void initMap() {
95     ((SupportMapFragment) getSupportFragmentManager().
96     findFragmentById(R.id.map2)).getMapAsync(new

```

```

91 OnMapReadyCallback() {
92     @Override
93     public void onMapReady(GoogleMap googleMap) {
94         client = new AsyncHttpClient();
95         mMap = googleMap;
96         mMap.getUiSettings().
setZoomControlsEnabled(true);
97         mMap.animateCamera(CameraUpdateFactory.
newLatLngZoom(new LatLng(-7.7752787,110.3848264), 12));
98
99         if (ActivityCompat.checkSelfPermission(
getApplicationContext(), android.Manifest.permission.
ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
&& ActivityCompat.checkSelfPermission(
getApplicationContext(), android.Manifest.permission.
ACCESS_COARSE_LOCATION) != PackageManager.
PERMISSION_GRANTED) {
100             // TODO: Consider calling
101             //     ActivityCompat#
requestPermissions
102             // here to request the missing
permissions, and then overriding
103             //     public void
onRequestPermissionsResult(int requestCode, String[]
permissions,
104             //                                     int[]
grantResults)
105             // to handle the case where the user
grants the permission. See the documentation
106             // for ActivityCompat#
requestPermissions for more details.
107             return;
108         }
109         mMap.setMyLocationEnabled(true);
110     }
111 });
112 }
113
114 boolean isLoadLocation = false;
115
116 private void GetLocationBy(LatLng latLng) {
117     String url = "https://maps.googleapis.com/maps/api
/geocode/json" +
118         "?latlng=" + latLng.latitude + "," +
latLng.longitude +
119         "&key=" + "
AIzaSyBPWWTZcnHwIfe8c_gLJkaLm8WGdJnLS7Y";
120
121     if(!isLoadLocation){
122         isLoadLocation = true;
123         client.get(url, new AsyncHttpResponseHandler()
{
124             @Override
125             public void onSuccess(int statusCode,

```



```
125 Header[] headers, byte[] responseBody) {
126     try{
127         String responseString = new String
(responseBody, "UTF-8");
128         JSONObject responseJsonObject =
new JSONObject(responseString);
129
130         JSONArray resultJsonArray =
responseJsonObject.getJSONArray("results");
131         JSONObject firstResult =
resultJsonArray.getJSONObject(0);
132         Log.d(TAG, firstResult.getString("
formatted_address"));
133         tvDestination.setText(firstResult.
getString("formatted_address"));
134
135         }catch (Exception e){
136             Log.d(TAG, e.getMessage());
137         }
138         isLoadingLocation=false;
139     }
140
141     @Override
142     public void onFailure(int statusCode,
Header[] headers, byte[] responseBody, Throwable error) {
143         Log.d(TAG, error.getMessage());
144         isLoadingLocation=false;
145     }
146     });
147 }
148 }
149 }
```

```
1 package aplikasiku.navwithdirectionlib;
2
3 import android.app.ProgressDialog;
4 import android.content.Context;
5 import android.location.Address;
6 import android.location.Geocoder;
7 import android.location.Location;
8 import android.location.LocationListener;
9 import android.location.LocationManager;
10 import android.support.v7.app.AppCompatActivity;
11 import android.os.Bundle;
12 import android.text.TextUtils;
13 import android.view.View;
14 import android.widget.Button;
15 import android.widget.EditText;
16 import android.widget.Toast;
17
18 import com.google.firebase.database.DatabaseError;
19 import com.google.firebase.database.DatabaseReference;
20 import com.google.firebase.database.FirebaseDatabase;
21
22 import java.util.List;
23 import java.util.Locale;
24
25 import aplikasiku.navwithdirectionlib.FDBMobel.Hospital;
26
27 public class InserterHospitalActivity extends
AppCompatActivity {
28
29     EditText editTextRSname, editTextRSloc, editTextRSlat,
editTextRSlong;
30     Button butGPS, butInsert;
31
32     GPS gps;
33
34     FirebaseDatabase mFirebaseDatabase;
35
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.activity_inserter_hospital)
;
40
41         init();
42     }
43
44     private void init() {
45         mFirebaseDatabase = FirebaseDatabase.getInstance();
46
47         editTextRSname = (EditText) findViewById(R.id.
editTextRSname);
48         editTextRSloc = (EditText) findViewById(R.id.
editTextRSloc);
49         editTextRSlat = (EditText) findViewById(R.id.
editTextRSlat);
```

```

50     editTextRSlong = (EditText) findViewById(R.id.
editTextRSlong);
51
52     butGPS = (Button) findViewById(R.id.
buttonInsertRSgetGPS);
53     butInsert = (Button) findViewById(R.id.
buttonInsertRSok);
54
55     butGPS.setOnClickListener(new View.OnClickListener
() {
56         @Override
57         public void onClick(View v) {
58             getGPSLoc();
59         }
60     });
61
62     butInsert.setOnClickListener(new View.
OnClickListener() {
63         @Override
64         public void onClick(View v) {
65             if(inputIsComplete()){
66                 insertFirebaseRS();
67             }else{
68                 Toast.makeText(getApplicationContext()
, "Please complete all input column", Toast.LENGTH_SHORT).
show();
69             }
70         }
71     });
72 }
73
74     private void insertFirebaseRS(){
75         final ProgressDialog pdDialog = ProgressDialog.
show(InserterHospitalActivity.this, "Insert RS", "Please
wait...", false, false);
76         mFirebaseDatabase.getReference("hospital").push().
setValue(
77             new Hospital(editTextRSname.getText().
toString(),
78                 editTextRSloc.getText().toString()
,
79                 "imageurl.com",
80                 Double.parseDouble(editTextRSlat.
getText().toString()),
81                 Double.parseDouble(editTextRSlong.
getText().toString()), new DatabaseReference.
CompletionListener() {
82             @Override
83             public void onComplete(DatabaseError
databaseError, DatabaseReference databaseReference) {
84                 if(databaseError==null){
85                     Toast.makeText(
getApplicationContext(), "Insert RS Ok", Toast.
LENGTH_SHORT).show();
86                     pdDialog.dismiss();

```

```
87         clearInput();
88     }
89     else{
90         Toast.makeText(
getApplicationContext(), "Insert RS Fail", Toast.
LENGTH_SHORT).show();
91     }
92 }
93 });
94 }
95
96 private boolean inputIsComplete(){
97     boolean state = true;
98     if(TextUtils.isEmpty(editTextRSname.getText())){
99         state = false;
100    }
101    else if(TextUtils.isEmpty(editTextRSloc.getText()
)}{
102        state = false;
103    }
104    else if(TextUtils.isEmpty(editTextRSlat.getText()
)}{
105        state = false;
106    }
107    else if(TextUtils.isEmpty(editTextRSlong.getText(
))){
108        state = false;
109    }
110
111    if(!TextUtils.isEmpty(editTextRSlat.getText())){
112        try{Double.parseDouble(editTextRSlat.getText()
.toString());}
113        catch(Exception e){
114            state = false;
115            Toast.makeText(getApplicationContext(), "
Latitude should be double value", Toast.LENGTH_SHORT).show
();
116        }
117    }
118
119    if(!TextUtils.isEmpty(editTextRSlong.getText())){
120        try{Double.parseDouble(editTextRSlong.getText(
.toString());}
121        catch(Exception e){
122            state = false;
123            Toast.makeText(getApplicationContext(), "
Longitude should be double value", Toast.LENGTH_SHORT).
show();
124        }
125    }
126
127    return state;
128 }
129
130 private void clearInput(){
```

```

131         editTextRSname.setText("");editTextRSname.
clearFocus();
132         editTextRSloc.setText("");editTextRSloc.clearFocus
();
133         editTextRSlat.setText("");editTextRSlat.clearFocus
();
134         editTextRSlong.setText("");editTextRSlong.
clearFocus();
135     }
136
137     private void getGPSLoc() {
138         gps = new InserterHospitalActivity.GPS();
139         if (gps.isGpsEnable()) {
140             gps.requestNewLocation();
141         } else {
142             Toast.makeText(getApplicationContext(), "GPS
feature not detected / available.", Toast.LENGTH_SHORT).
show();
143         }
144     }
145
146     private class GPS {
147         String gpsJalan, gpsDesa, gpsKecamatan,
gpsKabupaten, gpsProvinsi, gpsNegara, alamat;
148         Double latitude, longitude;
149         private LocationManager locationManager = null;
150         private LocationListener locationListener = null;
151         ProgressDialog pD1;
152
153         public GPS() {
154             locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
155         }
156
157         public void requestNewLocation() {
158             locationListener = new
InserterHospitalActivity.GPS.MyLocationListener();
159             locationManager.requestLocationUpdates(
LocationManager.GPS_PROVIDER, 5000, 10, locationListener);
160             pD1 = ProgressDialog.show(
InserterHospitalActivity.this, "Mencari Lokasi Anda", "
Harap tunggu...", true, false);
161         }
162
163         public boolean isGpsEnable(){
164             if(locationManager.isProviderEnabled(
LocationManager.GPS_PROVIDER)){ return true; }
165             else{ return false; }
166         }
167
168         private class MyLocationListener implements
LocationListener{
169             @Override
170             public void onLocationChanged(Location
location) {

```

```

171         //get long and lat
172         longitude = location.getLongitude();
173         latitude = location.getLatitude();
174
175         //get city name
176         Geocoder geo = new Geocoder(
177     applicationContext, Locale.getDefault());
178         List<Address> address;
179         try{
180             address = geo.getFromLocation(location
181     .getLatitude(), location.getLongitude(), 1);
182             if(address.size()>0){
183                 gpsJalan = address.get(0).
184     getAddressLine(0);
185                 gpsDesa = address.get(0).
186     getSubLocality();
187                 gpsKecamatan = address.get(0).
188     getLocality();
189                 gpsKabupaten = address.get(0).
190     getSubAdminArea();
191                 gpsProvinsi = address.get(0).
192     getAdminArea();
193                 gpsNegara = address.get(0).
194     getCountryName();
195                 alamat = gpsJalan + ", " + gpsDesa
196     + ", " + gpsKecamatan + ". ";
197                 //alamat+= gpsKabupaten + ", " +
198     gpsProvinsi + ", " + gpsNegara;
199                 pD1.dismiss();
200                 editTextRSloc.setText(alamat);
201                 editTextRSlat.setText(latitude.
202     toString());
203                 editTextRSlong.setText (longitude.
204     toString());
205                 locationManager.removeUpdates(
206     locationListener);
207             }
208         }catch(Exception e){
209             String error = e.getMessage();
210             Toast.makeText(getApplicationContext()
211     , error, Toast.LENGTH_SHORT).show();
212         }
213     }
214
215     @Override
216     public void onProviderEnabled(String provider)
217     {
218
219     }
220
221     @Override

```

```
211         public void onProviderDisabled(String provider
212     ) {
213         }
214
215         @Override
216         public void onStatusChanged(String provider,
217     int status, Bundle extras) {
218         }
219     }
220 }
221 }
222
```

```
1 package aplikasiku.navwithdirectionlib.GPS;
2
3 import com.google.android.gms.maps.model.LatLng;
4
5 /**
6  * Created by muiezhimura on 10/9/17.
7  */
8
9 public class LatLngLoc {
10     LatLng Location;
11     String NameLoc;
12     String TLkey;
13
14     public LatLngLoc(LatLng location, String nameLoc) {
15         Location = location;
16         NameLoc = nameLoc;
17     }
18
19     public LatLngLoc(LatLng location, String nameLoc,
20 String tlkey) {
21         Location = location;
22         NameLoc = nameLoc;
23         TLkey = tlkey;
24     }
25
26     public LatLng getLocation() {
27         return Location;
28     }
29
30     public String getNameLoc() {
31         return NameLoc;
32     }
33
34     public String getTLkey(){
35         return TLkey;
36     }
37 }
```



```
1 package aplikasiku.navwithdirectionlib.GPS;
2
3 import android.app.Service;
4 import android.content.Intent;
5 import android.location.Location;
6 import android.location.LocationListener;
7 import android.location.LocationManager;
8 import android.os.Bundle;
9 import android.os.Handler;
10 import android.os.IBinder;
11 import android.support.annotation.Nullable;
12 import android.util.Log;
13 import android.widget.Toast;
14
15 import java.util.Timer;
16 import java.util.TimerTask;
17
18 /**
19  * Created by deepshikha on 24/11/16.
20  */
21
22 public class GoogleService extends Service implements
LocationListener {
23
24     boolean isGPSEnable = false;
25     boolean isNetworkEnable = false;
26     double latitude, longitude;
27     LocationManager locationManager;
28     Location location;
29     private Handler mHandler = new Handler();
30     private Timer mTimer = null;
31     long notify_interval = 1000;
32     public static String str_receiver = "servicetutorial.
service.receiver";
33     Intent intent;
34
35     public GoogleService() {
36
37     }
38
39     @Nullable
40     @Override
41     public IBinder onBind(Intent intent) {
42         return null;
43     }
44
45     @Override
46     public void onCreate() {
47         super.onCreate();
48
49         mTimer = new Timer();
50         mTimer.schedule(new TimerTaskToGetLocation(), 5,
notify_interval);
51         intent = new Intent(str_receiver);
52         // fn_getlocation();
```

```
53     Log.e("gpsService", "create");
54 }
55
56 @Override
57 public void onLocationChanged(Location location) {
58     intent.putExtra("latitude", location.getLatitude()+
59     "");
60     intent.putExtra("longitude", location.getLongitude(
61     )+ "");
62     sendBroadcast(intent);
63 }
64
65 @Override
66 public void onStatusChanged(String provider, int
67 status, Bundle extras) {
68 }
69
70 @Override
71 public void onProviderEnabled(String provider) {
72 }
73
74 @Override
75 public void onProviderDisabled(String provider) {
76 }
77 }
78
79 private void fn_getlocation(){
80     locationManager = (LocationManager)
81     getApplicationContext().getSystemService(LOCATION_SERVICE)
82     ;
83     isGPSEnable = locationManager.isProviderEnabled(
84     locationManager.GPS_PROVIDER);
85     isNetworkEnable = locationManager.
86     isProviderEnabled(LocationManager.NETWORK_PROVIDER);
87
88     if (!isGPSEnable && !isNetworkEnable){
89         Log.e("gpsnetwork", "noEnable");
90     }else {
91         Log.e("gpsnetwork", "Enable");
92         if (isNetworkEnable){
93             Log.e("isNetworkEnable", "true");
94             location = null;
95             locationManager.requestLocationUpdates(
96             locationManager.NETWORK_PROVIDER, 1000, 0, this);
97             if (locationManager!=null){
98                 location = locationManager.
99                 getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
100                 if (location!=null){
101                     Log.e("latitude", location.
102                     getLatitude()+ "");
103                     Log.e("longitude", location.
```

```

97  getLongitude()+"");
98
99          latitude = location.getLatitude();
100         longitude = location.getLongitude(
101     );
102         fn_update(location);
103     }
104 }
105
106     if (isGPSEnable){
107         Log.e("isGPSEnable","true");
108         location = null;
109         locationManager.requestLocationUpdates(
110     locationManager.GPS_PROVIDER,1000,0,this);
111         if (locationManager!=null){
112             location = locationManager.
113     getLastKnownLocation(locationManager.GPS_PROVIDER);
114             if (location!=null){
115                 Log.e("latitude",location.
116     getLatitude()+"");
117                 Log.e("longitude",location.
118     getLongitude()+"");
119                 latitude = location.getLatitude();
120                 longitude = location.getLongitude(
121     );
122                 fn_update(location);
123             }
124         }
125     }
126 }
127
128     private class TimerTaskToGetLocation extends TimerTask
129     {
130         @Override
131         public void run() {
132             mHandler.post(new Runnable() {
133                 @Override
134                 public void run() {
135                     fn_getlocation();
136                 }
137             });
138         }
139     }
140
141     private void fn_update(Location location){
142         intent.putExtra("latutide",location.getLatitude()+
143     "");
144         intent.putExtra("longitude",location.getLongitude(
145     )+"");
146         sendBroadcast(intent);
147     }
148 }

```

```
1 package aplikasiku.navwithdirectionlib.GPS;
2
3 import com.google.android.gms.maps.model.LatLng;
4 import com.google.maps.android.SphericalUtil;
5
6 import java.util.ArrayList;
7
8 /**
9  * Created by muiezhimura on 10/9/17.
10 */
11
12 public class SortTLInRoute {
13     LatLng Start;
14     ArrayList<LatLngLoc> TLInRoute;
15     ArrayList<LatLngLoc> SortTLInRoute;
16
17     public SortTLInRoute(LatLng start, ArrayList<LatLngLoc>
18 TLInRoute) {
19         Start = start;
20         this.TLInRoute = TLInRoute;
21     }
22
23     public ArrayList<LatLngLoc> getSortTLInRoute() {
24         ArrayList<LatLngLoc> TempArray = new ArrayList<
25 LatLngLoc>();
26         LatLngLoc temp;
27
28         TempArray = TLInRoute;
29         if (TempArray.size()>1) {
30             for (int i = 0; i < TempArray.size() - 1; i++)
31             {
32                 for (int j = i +1; j < TempArray.size(); j
33 +++) {
34                     if (SphericalUtil.
35 computeDistanceBetween(Start, TempArray.get(i).getLocation(
36 )) > SphericalUtil.computeDistanceBetween(Start, TempArray.
37 get(j).getLocation())) {
38                         temp = TempArray.get(i);
39                         TempArray.set(i, TempArray.get(j));
40                         TempArray.set(j, temp);
41                     }
42                 }
43             }
44         }
45         return TempArray;
46     }
47 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 import com.google.firebase.database.IgnoreExtraProperties;
4
5 /**
6  * Created by Zulfa_K on 29-05-2017.
7  */
8
9 @IgnoreExtraProperties
10 public class Area {
11     public String area_name;
12
13     public Area(){}
14     public Area(String area_name){
15         this.area_name = area_name;
16     }
17 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 import com.google.firebase.database.Exclude;
4 import com.google.firebase.database.IgnoreExtraProperties;
5
6 import java.util.HashMap;
7 import java.util.Map;
8
9 /**
10  * Created by Zulfa_K on 22-05-2017.
11  */
12
13 @IgnoreExtraProperties
14 public class User {
15     public String name,status;
16
17     public User(){}
18     public User(String name,String status){
19         this.name = name;
20         this.status = status;
21     }
22
23     @Exclude
24     public Map<String, Object> toMap() {
25         HashMap<String, Object> result = new HashMap<>();
26         result.put("name", name);
27         result.put("status", status);
28         return result;
29     }
30 }
31
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 28-08-2017.
5  */
6
7 public class Route {
8     public String driver;
9     public String route_list;
10    public LogPerjalanan log;
11
12    public Route(){}
13    public Route(String driver, String route_list,
14    LogPerjalanan log){
15        this.driver = driver;
16        this.route_list = route_list;
17        this.log = log;
18    }
19 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 23-04-2017.
5  */
6 public class Hospital {
7     public String RsName,RsLoc,RsImageUrl;
8     public double RsLat, RsLong;
9
10    public Hospital(){}
11    public Hospital(String rname, String rsimageurl,
12    String rsloc, double rslat, double rslong){
13        RsName = rname;
14        RsLoc = rsloc;
15        RsImageUrl = rsimageurl;
16        RsLat = rslat;
17        RsLong = rslong;
18    }
19 }
```



```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 03-10-2017.
5  */
6
7 public class Requester {
8     //public String idUser; be firebase requester uid
9     public String type;
10    public int antrian;
11
12    public Requester(){
13        //this.idUser = "";
14        this.type = "";
15        this.antrian = 0;
16    }
17    public Requester(String type, int antrian){
18        //this.idUser = idUser;
19        this.type = type;
20        this.antrian = antrian;
21    }
22 }
23 }
24 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 20-06-2017.
5  */
6
7 public class NewSkemaTL {
8     public String TLname, TLloc;
9     public int TLcount;
10    public double TLlat, TLlong;
11    public NewSkemaTLpos TLpos;
12    public boolean TLstate;
13    public boolean TLplaysound;
14
15    public NewSkemaTL(){
16        TLname = "";
17        TLloc = "";
18        TLcount = 0;
19        TLlat = 0.0;
20        TLlong = 0.0;
21        TLpos = null;
22        TLstate = false;
23        TLplaysound = false;
24    }
25    public NewSkemaTL(String tlname, String tlloc, double
    tllat, double tllong, int tlcout, boolean tlstate, boolean
    tlplaysound, NewSkemaTLpos tlpos){
26        TLname = tlname;
27        TLloc = tlloc;
28        TLlat = tllat;
29        TLlong = tllong;
30        TLcount = tlcout;
31        TLpos=tlpos;
32        TLstate = tlstate;
33    }
34 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 import com.google.firebase.database.IgnoreExtraProperties;
4
5 /**
6  * Created by Zulfa_K on 29-05-2017.
7  */
8
9 @IgnoreExtraProperties
10 public class TLCondition {
11     public String TLpos;
12     public boolean TLstat;
13
14     public TLCondition(){}
15     public TLCondition(String tlpos, boolean tlstat){
16         TLpos = tlpos;
17         TLstat = tlstat;
18     }
19 }
20
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 import com.google.firebase.database.IgnoreExtraProperties;
4
5 /**
6  * Created by Zulfa_K on 29-05-2017.
7  */
8
9 @IgnoreExtraProperties
10 public class TrafficLight {
11     public String TLname, TLloc;
12     public int TLcount;
13     public double TLlat, TLlong;
14     public boolean TLisEmergency;
15
16
17     public TrafficLight(){}
18     public TrafficLight(String tlname, String tlloc, double
19     tllat, double tllong, int tlcout, boolean isEmergency){
20         TLname = tlname;
21         TLloc = tlloc;
22         TLlat = tllat;
23         TLlong = tllong;
24         TLcount = tlcout;
25         TLisEmergency = isEmergency;
26     }
27 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 28-08-2017.
5  */
6
7 public class LogPerjalanan {
8     public String alamat;
9     public Double latitude;
10    public Double longitude;
11
12    public LogPerjalanan(){}
13    public LogPerjalanan(String alamat, Double latitude,
14    Double longitude){
15        this.alamat = alamat;
16        this.latitude = latitude;
17        this.longitude = longitude;
18    }
19 }
20
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 20-06-2017.
5  */
6
7 public class NewSkemaTLpos {
8     public NewSkemaTLpoint TLu,TLs,TLt,TLb;
9
10    public NewSkemaTLpos(){
11        TLu = null;
12        TLs = null;
13        TLt = null;
14        TLb = null;
15    }
16    public NewSkemaTLpos(NewSkemaTLpoint tlb,
17        NewSkemaTLpoint tlt, NewSkemaTLpoint tlu, NewSkemaTLpoint
18        tls){
19        TLu = tlu;
20        TLs = tls;
21        TLt = tlt;
22        TLb = tlb;
23    }
24 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 20-06-2017.
5  */
6
7 public class NewSkemaRequest {
8     public boolean adaRequest;
9     public double jarakAcuan;
10    public long useNumber;
11    //public Requester perequest; push in firebase
12
13    public NewSkemaRequest(){
14        adaRequest = false;
15        jarakAcuan = 0;
16        useNumber = 0;
17    }
18    public NewSkemaRequest(boolean adaRequest, long jrk,
19    long useNumber){
20        this.adaRequest = adaRequest;
21        jarakAcuan = jrk;
22        this.useNumber = useNumber;
23        //perequest = new Requester();
24    }
25 }
```

```
1 package aplikasiku.navwithdirectionlib.FDBMobel;
2
3 /**
4  * Created by Zulfa_K on 20-06-2017.
5  */
6
7 public class NewSkemaTLpoint {
8     public boolean isEmergency;
9     public boolean stat;
10    public String pos;
11    public NewSkemaRequest request;
12    public String rlamp = "";
13    public String ulamp = "";
14
15    public NewSkemaTLpoint(){
16        request = null;
17        isEmergency = false;
18        stat = false;
19        pos = "";
20        rlamp = "";
21        ulamp = "";
22    }
23    public NewSkemaTLpoint(String pos, boolean isEmergency)
24    {
25        request = new NewSkemaRequest();
26        this.isEmergency = isEmergency;
27        stat = true;
28        this.pos = pos;
29        rlamp = "green";
30        ulamp = "green";
31    }
32 }
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:paddingTop="15dp"
7     tools:context="aplikasiku.navwithdirectionlib.
  AdminActivity">
8
9     <LinearLayout
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:orientation="vertical"
13        android:id="@+id/layoutadmintlarea">
14        <TextView
15            android:layout_width="match_parent"
16            android:layout_height="wrap_content"
17            android:text="TL Area"
18            android:layout_marginLeft="4dp"
19            android:textAppearance="@style/TextAppearance.
  AppCompat.Medium"/>
20        <Spinner
21            android:id="@+id/spinnerAdminTLarea"
22            android:layout_width="match_parent"
23            android:layout_height="wrap_content"
24            android:entries="@array/itemArea"/>
25        <View
26            android:layout_width="match_parent"
27            android:layout_height="2dp"
28            android:layout_marginRight="4dp"
29            android:layout_marginLeft="4dp"
30            android:background="@color/divider"/>
31    </LinearLayout>
32
33    <android.support.v7.widget.RecyclerView
34        android:id="@+id/recyclerview_admin_tllist"
35        android:layout_width="match_parent"
36        android:layout_height="match_parent"
37        android:layout_below="@id/layoutadmintlarea"
38        android:layout_marginBottom="160dp"/>
39
40    <Button
41        android:id="@+id/buttonAdminKontrolLampu"
42        android:layout_width="match_parent"
43        android:layout_height="wrap_content"
44        android:layout_marginTop="-150dp"
45        android:layout_below="@id/recyclerview_admin_tllist"
46        android:text="Kontroler Lampu" />
47
48    <Button
49        android:id="@+id/buttonAdminInsertTL"
50        android:layout_width="match_parent"
51        android:layout_height="wrap_content"
```

```
52         android:layout_below="@id/buttonAdminKontrolLampu"
53         android:text="Inserter TL" />
54
55     <Button
56         android:id="@+id/buttonAdminInsertRS"
57         android:layout_width="match_parent"
58         android:layout_height="wrap_content"
59         android:layout_below="@id/buttonAdminInsertTL"
60         android:text="Inserter RS" />
61 </RelativeLayout>
62
```

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     android:paddingBottom="@dimen/activity_vertical_margin"
7     android:paddingLeft="@dimen/activity_horizontal_margin"
8     android:paddingRight="@dimen/activity_horizontal_margin"
9     "
10    android:paddingTop="@dimen/activity_vertical_margin"
11    tools:context="aplikasiku.navwithdirectionlib.
  LoginActivity">
12    <RelativeLayout
13        android:layout_width="match_parent"
14        android:layout_height="match_parent">
15        <!-- Login progress -->
16        <ProgressBar
17            android:id="@+id/login_progress"
18            style="?android:attr/progressBarStyleLarge"
19            android:layout_width="wrap_content"
20            android:layout_height="wrap_content"
21            android:layout_marginTop="8dp"
22            android:visibility="gone"
23            android:layout_centerVertical="true"
24            android:layout_centerHorizontal="true"/>
25    </RelativeLayout>
26
27    <ScrollView
28        android:id="@+id/login_form"
29        android:layout_width="match_parent"
30        android:layout_height="match_parent">
31
32        <LinearLayout
33            android:id="@+id/email_login_form"
34            android:layout_width="match_parent"
35            android:layout_height="wrap_content"
36            android:orientation="vertical">
37
38            <ImageView
39                android:layout_width="225dp"
40                android:layout_height="225dp"
41                android:src="@drawable/app_withname"
42                android:layout_marginTop="35dp"
43                android:layout_gravity="center_horizontal"
44            />
45
46            <!-- space after logo -->
47            <ImageView
48                android:layout_width="wrap_content"
49                android:layout_height="wrap_content"
50                android:layout_marginTop="45dp"/>
51
52            <android.support.design.widget.TextInputLayout
```

```
52         android:layout_width="match_parent"
53         android:layout_height="wrap_content">
54
55         <AutoCompleteTextView
56             android:id="@+id/email"
57             android:layout_width="match_parent"
58             android:layout_height="wrap_content"
59             android:hint="@string/prompt_email"
60             android:inputType="textEmailAddress"
61             android:maxLines="1"
62             android:layout_marginLeft="15dp"
63             android:layout_marginRight="15dp" />
64
65     </android.support.design.widget.
TextInputLayout>
66
67     <android.support.design.widget.TextInputLayout
68         android:layout_width="match_parent"
69         android:layout_height="wrap_content">
70
71         <EditText
72             android:id="@+id/password"
73             android:layout_width="match_parent"
74             android:layout_height="wrap_content"
75             android:hint="Password"
76             android:imeActionId="@+id/login"
77             android:imeActionLabel="@string/
action_sign_in_short"
78             android:imeOptions="actionUnspecified"
79             android:inputType="textPassword"
80             android:layout_marginLeft="15dp"
81             android:layout_marginRight="15dp"
82             android:maxLines="1"
83             android:singleLine="true" />
84
85     </android.support.design.widget.
TextInputLayout>
86
87     <Button
88         android:id="@+id/email_sign_in_button"
89         style="?android:textAppearanceSmall"
90         android:layout_width="match_parent"
91         android:layout_height="wrap_content"
92         android:layout_marginTop="30dp"
93         android:layout_marginLeft="20dp"
94         android:layout_marginRight="20dp"
95         android:text="Sign In"
96         android:textStyle="bold" />
97
98     <TextView
99         android:id="@+id/textViewSignup"
100        android:layout_width="match_parent"
101        android:layout_height="wrap_content"
102        android:layout_marginTop="25dp"
103        android:gravity="center"
```

```
104         android:visibility="visible"
105         android:text="Dont have an account? Sign
        up!" />
106
107     </LinearLayout>
108 </ScrollView>
109 </RelativeLayout>
110
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     android:orientation="vertical" android:layout_width="
match_parent"
4     android:layout_height="match_parent"
5     android:padding="5dp"
6     android:gravity="center">
7
8     <android.support.v7.widget.CardView
9         xmlns:card_view="http://schemas.android.com/apk/res
-auto"
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:id="@+id/listcardview"
13        card_view:cardElevation="2dp"
14        card_view:cardCornerRadius="5dp"
15        card_view:cardUseCompatPadding="true">
16
17        <LinearLayout
18            android:layout_width="match_parent"
19            android:layout_height="wrap_content"
20            android:orientation="vertical"
21            android:gravity="center_vertical"
22            android:padding="3dp">
23
24            <ImageView
25                android:layout_width="match_parent"
26                android:layout_height="150dp"
27                android:id="@+id/imageViewRSImage"
28                android:contentDescription="Hospital Image"
29                android:scaleType="centerCrop"
30                android:layout_gravity="center_horizontal"
31            />
32
33            <TextView
34                android:layout_width="wrap_content"
35                android:layout_height="wrap_content"
36                android:textAppearance="?android:attr/
textAppearanceLarge"
37                android:text="Rumah Sakit name"
38                android:id="@+id/textViewRSName" />
39
40            <TextView
41                android:layout_width="wrap_content"
42                android:layout_height="wrap_content"
43                android:textAppearance="?android:attr/
textAppearanceSmall"
44                android:text="rs location"
45                android:id="@+id/textViewRSLoc" />
46        </LinearLayout>
47    </android.support.v7.widget.CardView>
48
49 </LinearLayout>
```

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical"
6     android:paddingBottom="@dimen/activity_vertical_margin"
7     android:paddingLeft="@dimen/activity_horizontal_margin"
8     android:paddingRight="@dimen/activity_horizontal_margin"
9     android:paddingTop="@dimen/activity_vertical_margin"
10    tools:context="aplikasiku.navwithdirectionlib.
  SignupActivity">
11
12    <RelativeLayout
13        android:layout_width="match_parent"
14        android:layout_height="match_parent">
15        <!-- Login progress v -->
16        <ProgressBar
17            android:id="@+id/login_progress"
18            style="?android:attr/progressBarStyleLarge"
19            android:layout_width="wrap_content"
20            android:layout_height="wrap_content"
21            android:layout_marginBottom="8dp"
22            android:visibility="gone"
23            android:layout_gravity="center"
24            android:layout_centerVertical="true"
25            android:layout_centerHorizontal="true" />
26    </RelativeLayout>
27
28    <ScrollView
29        android:id="@+id/login_form"
30        android:layout_width="match_parent"
31        android:layout_height="match_parent">
32
33        <LinearLayout
34            android:id="@+id/email_login_form"
35            android:layout_width="match_parent"
36            android:layout_height="wrap_content"
37            android:orientation="vertical">
38
39            <ImageView
40                android:layout_width="225dp"
41                android:layout_height="225dp"
42                android:src="@drawable/app_withname"
43                android:layout_marginTop="35dp"
44                android:layout_gravity="center_horizontal"
45            />
46
47            <!-- space after logo -->
48            <ImageView
49                android:layout_width="wrap_content"
50                android:layout_height="wrap_content"
51                android:layout_marginTop="45dp" />
```

```
52         <android.support.design.widget.TextInputLayout
53             android:layout_width="match_parent"
54             android:layout_height="wrap_content">
55
56             <AutoCompleteTextView
57                 android:id="@+id/email"
58                 android:layout_width="match_parent"
59                 android:layout_height="wrap_content"
60                 android:hint="@string/prompt_email"
61                 android:inputType="textEmailAddress"
62                 android:maxLines="1"
63                 android:layout_marginLeft="15dp"
64                 android:layout_marginRight="15dp"
65                 android:singleLine="true" />
66
67         </android.support.design.widget.
68 TextInputLayout>
69
70         <android.support.design.widget.TextInputLayout
71             android:layout_width="match_parent"
72             android:layout_height="wrap_content">
73
74             <EditText
75                 android:id="@+id/password"
76                 android:layout_width="match_parent"
77                 android:layout_height="wrap_content"
78                 android:hint="Password"
79                 android:imeActionId="@+id/login"
80                 android:imeActionLabel="@string/
81 action_sign_in_short"
82                 android:imeOptions="actionUnspecified"
83                 android:inputType="textPassword"
84                 android:layout_marginLeft="15dp"
85                 android:layout_marginRight="15dp"
86                 android:maxLines="1"
87                 android:singleLine="true" />
88
89         </android.support.design.widget.
90 TextInputLayout>
91
92         <Button
93             android:id="@+id/email_sign_in_button"
94             style="?android:textAppearanceSmall"
95             android:layout_width="match_parent"
96             android:layout_height="wrap_content"
97             android:layout_marginTop="25dp"
98             android:layout_marginLeft="20dp"
99             android:layout_marginRight="20dp"
100            android:text="Register New Account"
101            android:textStyle="bold" />
102
103         <TextView
104             android:layout_width="match_parent"
105             android:layout_height="wrap_content"
106             android:layout_marginTop="25dp"
```



```
104         android:gravity="center"
105         android:visibility="invisible"
106         android:text="Already have Account? Sign
      me!"/>
107
108     </LinearLayout>
109 </ScrollView>
110 </RelativeLayout>
111
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:paddingBottom="@dimen/activity_vertical_margin"
8     android:paddingLeft="@dimen/activity_horizontal_margin"
9     android:paddingRight="@dimen/activity_horizontal_margin"
10    "
11    android:paddingTop="@dimen/activity_vertical_margin"
12    tools:context="aplikasiku.navwithdirectionlib.hal_map">
13    <TextView
14        android:id="@+id/halmapinfo"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:gravity="center"
18        android:text="info"
19        android:textSize="18sp"
20        android:textStyle="bold" />
21
22    <fragment
23        android:id="@+id/map"
24        android:name="com.google.android.gms.maps.
25    SupportMapFragment"
26        android:layout_width="match_parent"
27        android:layout_height="match_parent"
28        android:layout_alignParentStart="true"
29        android:layout_alignParentTop="true"
30        android:layout_marginTop="10dp" />
31 </LinearLayout>
32
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools" android:
  layout_width="match_parent"
4     android:layout_height="match_parent" android:
  paddingLeft="@dimen/activity_horizontal_margin"
5     android:paddingRight="@dimen/activity_horizontal_margin"
  "
6     android:paddingTop="@dimen/activity_vertical_margin"
7     android:paddingBottom="@dimen/activity_vertical_margin"
  tools:context=".hal_home"
8     android:background="#FFE8E8">
9
10    <TextView
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="Data User."/>
14
15    <android.support.v7.widget.CardView xmlns:card_view="
  http://schemas.android.com/apk/res-auto"
16        android:id="@+id/layoutProfile"
17        android:layout_width="match_parent"
18        android:layout_height="wrap_content"
19        android:layout_marginTop="25dp"
20        card_view:cardCornerRadius="5dp"
21        card_view:cardElevation="2dp"
22        card_view:cardUseCompatPadding="true">
23
24        <LinearLayout
25            android:layout_width="match_parent"
26            android:layout_height="wrap_content"
27            android:layout_marginBottom="10dp"
28            android:layout_marginLeft="10dp"
29            android:layout_marginRight="10dp"
30            android:layout_marginTop="10dp"
31            android:orientation="horizontal">
32
33            <ImageView
34                android:layout_width="75dp"
35                android:layout_height="75dp"
36                android:src="@drawable/account_logo" />
37
38            <ScrollView
39                android:layout_width="match_parent"
40                android:layout_height="match_parent">
41
42                <LinearLayout
43                    android:layout_width="match_parent"
44                    android:layout_height="wrap_content"
45                    android:layout_marginLeft="10dp"
46                    android:gravity="center_vertical"
47                    android:orientation="vertical">
48
49                    <TextView
```

```

50         android:id="@+id/textViewIdLogin"
51         android:layout_width="wrap_content"
52         "
53         android:layout_height="
54         wrap_content"
55         android:text="identitas login" />
56     <TextView
57         android:id="@+id/
58         textViewPosisiLogin"
59         android:layout_width="wrap_content"
60         "
61         android:layout_height="
62         wrap_content"
63         android:text="posisi" />
64     <TextView
65         android:id="@+id/textViewLatLogin"
66         android:layout_width="wrap_content"
67         "
68         android:layout_height="
69         wrap_content"
70         android:text="Latitude" />
71     <TextView
72         android:id="@+id/textViewLongLogin
73         "
74         android:layout_width="wrap_content"
75         "
76         android:layout_height="
77         wrap_content"
78         android:text="Longitude" />
79     </LinearLayout>
80 </ScrollView>
81 </LinearLayout>
82 </android.support.v7.widget.CardView>
83 <TextView android:text="Data RS." android:layout_width
84 ="wrap_content"
85         android:layout_height="wrap_content"
86         android:id="@+id/textView"
87         android:layout_below="@id/layoutProfile"
88         android:layout_marginTop="10dp"/>
89     <android.support.v7.widget.RecyclerView
90         android:id="@+id/recyclerViewListRS"
91         android:layout_width="match_parent"
92         android:layout_height="match_parent"
93         android:layout_above="@+id/buttonKontrolerLampu"
94         android:layout_below="@+id/textView"
95         android:scrollbars="vertical" />
96     <Button
97         android:layout_width="match_parent"
98         android:layout_height="wrap_content"

```

```
94         android:text="Admin Page"
95         android:layout_alignParentEnd="true"
96         android:layout_alignParentBottom="true"
97         android:id="@+id/buttonKontrolerLampu"/>
98
99     <Button
100         android:layout_width="match_parent"
101         android:layout_height="wrap_content"
102         android:text="Set Destination"
103         android:layout_marginBottom="50dp"
104         android:layout_alignParentBottom="true"
105         android:id="@+id/butSetDestination"/>
106
107 </RelativeLayout>
108
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:padding="10dp"
8     tools:context="aplikasiku.navwithdirectionlib.
SkemaTLbaruActivity">
9
10
11     <TextView
12         android:id="@+id/textView3"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:text="TL Area"
16         android:layout_marginLeft="4dp"
17         android:textAppearance="@style/TextAppearance.
AppCompat.Medium"/>
18     <Spinner
19         android:id="@+id/spinnerTLarea"
20         android:layout_width="match_parent"
21         android:layout_height="wrap_content"
22         android:entries="@array/itemArea"/>
23     <View
24         android:layout_width="match_parent"
25         android:layout_height="2dp"
26         android:layout_marginRight="4dp"
27         android:layout_marginLeft="4dp"
28         android:background="@color/divider"/>
29
30     <!-- dummy item focusable first -->
31     <LinearLayout
32         android:focusable="true"
33         android:focusableInTouchMode="true"
34         android:layout_width="0px"
35         android:layout_height="0px"/>
36
37     <android.support.design.widget.TextInputLayout
38         android:layout_width="match_parent"
39         android:layout_height="wrap_content"
40         android:layout_marginTop="5dp">
41         <EditText
42             android:id="@+id/editTextTLname"
43             android:layout_width="match_parent"
44             android:layout_height="wrap_content"
45             android:hint="TL Name"
46             android:maxLines="1"
47             android:nextFocusUp="@id/editTextTLname"
48             android:nextFocusLeft="@id/editTextTLname" />
49     </android.support.design.widget.TextInputLayout>
50
51     <LinearLayout
52         android:layout_width="match_parent"
```

```
53     android:layout_height="wrap_content"
54     android:orientation="horizontal"
55     android:gravity="center">
56     <android.support.design.widget.TextInputLayout
57         android:layout_width="match_parent"
58         android:layout_height="wrap_content"
59         android:layout_marginRight="80dp">
60         <EditText
61             android:id="@+id/editTextTLloc"
62             android:layout_width="match_parent"
63             android:layout_height="wrap_content"
64             android:hint="TL Location" />
65     </android.support.design.widget.TextInputLayout>
66     <Button
67         android:id="@+id/buttonGetLocTL"
68         android:layout_width="wrap_content"
69         android:layout_height="wrap_content"
70         android:layout_marginLeft="-75dp"
71         android:text="GPS"/>
72 </LinearLayout>
73
74 <android.support.design.widget.TextInputLayout
75     android:layout_width="match_parent"
76     android:layout_height="wrap_content">
77     <EditText
78         android:id="@+id/editTextTLlat"
79         android:layout_width="match_parent"
80         android:layout_height="wrap_content"
81         android:hint="TL Latitude"
82         android:maxLines="1"/>
83 </android.support.design.widget.TextInputLayout>
84
85 <android.support.design.widget.TextInputLayout
86     android:layout_width="match_parent"
87     android:layout_height="wrap_content">
88     <EditText
89         android:id="@+id/editTextTLlong"
90         android:layout_width="match_parent"
91         android:layout_height="wrap_content"
92         android:hint="TL Longitude"
93         android:maxLines="1"/>
94 </android.support.design.widget.TextInputLayout>
95
96 <TextView
97     android:id="@+id/textView4"
98     android:layout_width="match_parent"
99     android:layout_height="wrap_content"
100    android:text="TL Count"
101    android:layout_marginTop="10dp"
102    android:layout_marginLeft="4dp"
103    android:textAppearance="@style/TextAppearance.
AppCompat.Medium"/>
104
105 <LinearLayout
106     android:layout_width="match_parent"
```

```
107     android:layout_height="wrap_content"
108     android:orientation="horizontal">
109     <CheckBox
110         android:id="@+id/checkBoxTLb"
111         android:layout_width="wrap_content"
112         android:layout_height="wrap_content"
113         android:text="TLb" />
114     <CheckBox
115         android:id="@+id/checkBoxTLs"
116         android:layout_width="wrap_content"
117         android:layout_height="wrap_content"
118         android:layout_marginLeft="5dp"
119         android:text="TLs" />
120     <CheckBox
121         android:id="@+id/checkBoxTLt"
122         android:layout_width="wrap_content"
123         android:layout_height="wrap_content"
124         android:layout_marginLeft="5dp"
125         android:text="TLt" />
126     <CheckBox
127         android:id="@+id/checkBoxTLu"
128         android:layout_width="wrap_content"
129         android:layout_height="wrap_content"
130         android:layout_marginLeft="5dp"
131         android:text="TLu" />
132 </LinearLayout>
133 <View
134     android:layout_width="match_parent"
135     android:layout_height="2dp"
136     android:layout_marginRight="4dp"
137     android:layout_marginLeft="4dp"
138     android:background="@color/divider"/>
139
140 <Button
141     android:id="@+id/buttonAddTL"
142     android:layout_width="match_parent"
143     android:layout_height="wrap_content"
144     android:text="Insert TL"
145     android:layout_marginTop="25dp"/>
146
147 </LinearLayout>
148
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     android:orientation="vertical" android:layout_width="
  match_parent"
4     android:layout_height="wrap_content"
5     android:gravity="center"
6     android:layout_marginTop="5dp"
7     android:padding="5dp">
8
9     <TextView
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:text="Nama TL"
13         android:id="@+id/textViewListTLName"/>
14     <TextView
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:text="Alamat TL"
18         android:id="@+id/textViewListTLAlamat"/>
19     <LinearLayout
20         android:layout_width="match_parent"
21         android:layout_height="wrap_content"
22         android:orientation="horizontal">
23         <TextView
24             android:layout_width="wrap_content"
25             android:layout_height="wrap_content"
26             android:text="Latitude TL"
27             android:id="@+id/textViewListTLLat"/>
28         <TextView
29             android:layout_width="wrap_content"
30             android:layout_height="wrap_content"
31             android:layout_marginLeft="10dp"
32             android:text="Longitude TL"
33             android:id="@+id/textViewListTLLong"/>
34     </LinearLayout>
35     <View
36         android:layout_width="match_parent"
37         android:layout_height="1dp"
38         android:background="@color/divider"/>
39
40 </LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     tools:context="aplikasiku.navwithdirectionlib.
  KontrolerLampuActivity">
7
8     <LinearLayout
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:orientation="vertical"
12        android:id="@+id/layoutkontrolertlarea">
13        <TextView
14            android:id="@+id/textView33"
15            android:layout_width="match_parent"
16            android:layout_height="wrap_content"
17            android:text="TL Area"
18            android:layout_marginLeft="4dp"
19            android:textAppearance="@style/TextAppearance.
  AppCompat.Medium"/>
20        <Spinner
21            android:id="@+id/spinnerKontrolerTLarea"
22            android:layout_width="match_parent"
23            android:layout_height="wrap_content"
24            android:entries="@array/itemArea"/>
25        <View
26            android:layout_width="match_parent"
27            android:layout_height="2dp"
28            android:layout_marginRight="4dp"
29            android:layout_marginLeft="4dp"
30            android:background="@color/divider"/>
31    </LinearLayout>
32
33    <android.support.v7.widget.RecyclerView
34        android:layout_width="match_parent"
35        android:layout_height="match_parent"
36        android:layout_marginBottom="280dp"
37        android:layout_marginTop="10dp"
38        android:layout_below="@id/layoutkontrolertlarea"
39        android:id="@+id/recyclerViewListLampuKontroler"/>
40
41    <LinearLayout
42        android:layout_width="match_parent"
43        android:layout_height="wrap_content"
44        android:orientation="vertical"
45        android:layout_marginTop="-270dp"
46        android:layout_centerHorizontal="true"
47        android:layout_below="@+id/
  recyclerViewListLampuKontroler">
48
49        <RelativeLayout
50            android:layout_width="match_parent"
51            android:layout_height="wrap_content"
```

```

52         android:padding="10dp">
53         <TextView
54             android:layout_width="wrap_content"
55             android:layout_height="wrap_content"
56             android:text="Kontroler Lampu"
57             android:textSize="18sp"
58             android:layout_centerVertical="true"/>
59         <RadioButton
60             android:layout_width="wrap_content"
61             android:layout_height="wrap_content"
62             android:text="Manual"
63             android:layout_toLeftOf="@+id/
radioKontrolerOtomatis"
64             android:id="@+id/radioKontrolerManual"
        />
65         <RadioButton
66             android:layout_width="wrap_content"
67             android:layout_height="wrap_content"
68             android:text="Otomatis"
69             android:layout_alignParentRight="true"
70             android:id="@+id/
radioKontrolerOtomatis"/>
71     </RelativeLayout>
72
73     <View
74         android:layout_width="match_parent"
75         android:layout_height="1dp"
76         android:background="#FFDFDDDD"/>
77
78     <RelativeLayout
79         android:layout_width="match_parent"
80         android:layout_height="wrap_content"
81         android:padding="5dp">
82         <TextView
83             android:layout_width="wrap_content"
84             android:layout_height="wrap_content"
85             android:text="TL barat"
86             android:layout_centerVertical="true"
87             android:layout_alignParentLeft="true"
        />
88
89     <LinearLayout
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:orientation="horizontal"
93         android:layout_toLeftOf="@+id/
switchLampuBarat"
94         android:layout_marginRight="10dp">
95         <RadioButton
96             android:layout_width="wrap_content
97             android:layout_height="
wrap_content"
98             android:text="R"
99             android:id="@+id/

```

```

99  radioLampuBaratRed"/>
100      <RadioButton
101          android:layout_width="wrap_content
102      "
103          android:layout_height="
104      wrap_content"
105          android:text="Y"
106          android:id="@+id/
107      radioLampuBaratYellow"/>
108      <RadioButton
109          android:layout_width="wrap_content
110      "
111          android:layout_height="
112      wrap_content"
113          android:text="G"
114          android:id="@+id/
115      radioLampuBaratGreen"/>
116      </LinearLayout>
117  </RelativeLayout>
118
119  <View
120      android:layout_width="match_parent"
121      android:layout_height="1dp"
122      android:background="#FFDFDDDD"/>
123
124  <RelativeLayout
125      android:layout_width="match_parent"
126      android:layout_height="wrap_content"
127      android:padding="5dp">
128      <TextView
129          android:layout_width="wrap_content"
130          android:layout_height="wrap_content"
131          android:text="TL timur"
132          android:layout_centerVertical="true"
133          android:layout_alignParentLeft="true"
134      />
135
136  <LinearLayout
137      android:layout_width="wrap_content"
138      android:layout_height="wrap_content"
139      android:orientation="horizontal"
140      android:layout_toLeftOf="@+id/
141      switchLampuTimur"
142      android:layout_marginRight="10dp">
143      <RadioButton
144          android:layout_width="wrap_content
145      "
146          android:layout_height="

```

```

143 wrap_content"
144         android:text="R"
145         android:id="@+id/
radioLampuTimurRed"/>
146         <RadioButton
147         android:layout_width="wrap_content
"
148         android:layout_height="
wrap_content"
149         android:text="Y"
150         android:id="@+id/
radioLampuTimurYellow"/>
151         <RadioButton
152         android:layout_width="wrap_content
"
153         android:layout_height="
wrap_content"
154         android:text="G"
155         android:id="@+id/
radioLampuTimurGreen"/>
156     </LinearLayout>
157
158     <Switch
159         android:id="@+id/switchLampuTimur"
160         android:layout_width="wrap_content"
161         android:layout_height="wrap_content"
162         android:layout_alignParentRight="true"
/>
163 </RelativeLayout>
164
165 <View
166     android:layout_width="match_parent"
167     android:layout_height="1dp"
168     android:background="#FFDFDDDD"/>
169
170 <RelativeLayout
171     android:layout_width="match_parent"
172     android:layout_height="wrap_content"
173     android:padding="5dp">
174     <TextView
175         android:layout_width="wrap_content"
176         android:layout_height="wrap_content"
177         android:text="TL utara"
178         android:layout_centerVertical="true"
179         android:layout_alignParentLeft="true"
/>
180
181     <LinearLayout
182         android:layout_width="wrap_content"
183         android:layout_height="wrap_content"
184         android:orientation="horizontal"
185         android:layout_toLeftOf="@+id/
switchLampuUtara"
186         android:layout_marginRight="10dp">
187     <RadioButton

```

```

188         android:layout_width="wrap_content
189         android:layout_height="
wrap_content"
190         android:text="R"
191         android:id="@+id/
radioLampuUtaraRed"/>
192         <RadioButton
193         android:layout_width="wrap_content
194         android:layout_height="
wrap_content"
195         android:text="Y"
196         android:id="@+id/
radioLampuUtaraYellow"/>
197         <RadioButton
198         android:layout_width="wrap_content
199         android:layout_height="
wrap_content"
200         android:text="G"
201         android:id="@+id/
radioLampuUtaraGreen"/>
202     </LinearLayout>
203
204     <Switch
205         android:id="@+id/switchLampuUtara"
206         android:layout_width="wrap_content"
207         android:layout_height="wrap_content"
208         android:layout_alignParentRight="true"
/>
209 </RelativeLayout>
210
211 <View
212     android:layout_width="match_parent"
213     android:layout_height="1dp"
214     android:background="#FFDFDDDD"/>
215
216 <RelativeLayout
217     android:layout_width="match_parent"
218     android:layout_height="wrap_content"
219     android:padding="5dp">
220     <TextView
221         android:layout_width="wrap_content"
222         android:layout_height="wrap_content"
223         android:text="TL selatan"
224         android:layout_centerVertical="true"
225         android:layout_alignParentLeft="true"
/>
226
227     <LinearLayout
228         android:layout_width="wrap_content"
229         android:layout_height="wrap_content"
230         android:orientation="horizontal"
231         android:layout_toLeftOf="@+id/

```

```

231 switchLampuSelatan"
232         android:layout_marginRight="10dp">
233         <RadioButton
234         "
235         android:layout_height="
236         wrap_content"
237         android:text="R"
238         android:id="@+id/
239         radioLampuSelatanRed"/>
240         <RadioButton
241         "
242         android:layout_height="
243         wrap_content"
244         android:text="Y"
245         android:id="@+id/
246         radioLampuSelatanYellow"/>
247         <RadioButton
248         "
249         android:layout_height="
250         wrap_content"
251         android:text="G"
252         android:id="@+id/
253         radioLampuSelatanGreen"/>
254         </LinearLayout>
255     />
256     </RelativeLayout>
257     <View
258     android:layout_width="match_parent"
259     android:layout_height="1dp"
260     android:background="#FFDFDDDD"/>
261     <RelativeLayout
262     android:layout_width="match_parent"
263     android:layout_height="wrap_content"
264     android:padding="5dp">
265     <TextView
266     android:layout_width="wrap_content"
267     android:layout_height="wrap_content"
268     android:text="TL playsound"
269     android:layout_centerVertical="true"
270     android:layout_alignParentLeft="true"
271     />
272     <Switch
273     android:id="@+id/switchTLplaysound"
274

```

```
275         android:layout_width="wrap_content"
276         android:layout_height="wrap_content"
277         android:layout_alignParentRight="true"
    />
278     </RelativeLayout>
279 </LinearLayout>
280
281 </RelativeLayout>
282
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     xmlns:app="http://schemas.android.com/apk/res-auto"
7     tools:context="aplikasiku.navwithdirectionlib.
  SetDestinationActivity">
8
9     <fragment
10         android:id="@+id/map2"
11         android:name="com.google.android.gms.maps.
  SupportMapFragment"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent" />
14
15     <ImageView
16         android:id="@+id/iv_center_location"
17         android:layout_width="25dp"
18         android:layout_height="25dp"
19         android:layout_centerHorizontal="true"
20         android:layout_centerVertical="true"
21         android:visibility="gone"
22         android:src="@drawable/ic_center_location"/>
23
24     <android.support.v7.widget.CardView
25         android:layout_width="match_parent"
26         android:layout_height="wrap_content"
27         android:layout_marginLeft="10dp"
28         android:layout_marginRight="10dp"
29         android:layout_marginTop="10dp">
30
31         <LinearLayout
32             android:layout_width="match_parent"
33             android:layout_height="wrap_content"
34             android:padding="10dp"
35             android:orientation="vertical">
36
37             <LinearLayout
38                 android:layout_width="match_parent"
39                 android:layout_height="wrap_content"
40                 android:orientation="horizontal"
41                 android:id="@+id/layout_origin_address">
42                 <ImageView
43                     android:layout_width="20dp"
44                     android:layout_height="20dp"
45                     android:layout_gravity="center_vertical"
46                     android:src="@drawable/marker_ambulance"
47                 />
48                 <TextView
49                     android:layout_marginLeft="10dp"
50                     android:layout_width="match_parent"
```

```
51         android:layout_height="wrap_content"
52         android:layout_gravity="
center_horizontal"
53         android:id="@+id/tv_origin_address"
54         android:text="Origin address..."/>
55     </LinearLayout>
56
57     <View
58     android:layout_width="match_parent"
59     android:layout_height="1dp"
60     android:background="#cfcfc9"
61     android:layout_marginTop="5dp"/>
62
63     <LinearLayout
64     android:layout_width="match_parent"
65     android:layout_height="wrap_content"
66     android:orientation="horizontal"
67     android:id="@+id/layout_dest_address"
68     android:layout_marginTop="10dp">
69         <ImageView
70         android:layout_width="20dp"
71         android:layout_height="20dp"
72         android:layout_gravity="
center_vertical"
73         android:src="@drawable/
ic_center_location"
74         />
75         <TextView
76         android:layout_marginLeft="10dp"
77         android:layout_width="match_parent"
78         android:layout_height="wrap_content"
79         android:layout_gravity="
center_horizontal"
80         android:id="@+id/tv_des_address"
81         android:text="Destination address..."
/>
82     </LinearLayout>
83
84     <View
85     android:layout_width="match_parent"
86     android:layout_height="1dp"
87     android:background="#cfcfc9"
88     android:layout_marginTop="5dp"/>
89
90     <LinearLayout
91     android:layout_width="match_parent"
92     android:layout_height="wrap_content"
93     android:orientation="horizontal"
94     android:id="@+id/layout_search_address"
95     android:layout_marginTop="5dp">
96         <fragment
97         android:id="@+id/
place_autocomplete_fragment"
98         android:layout_width="match_parent"
99         android:layout_height="40dp"
```

```
100         android:name="com.google.android.gms.  
location.places.ui.PlaceAutocompleteFragment" />  
101     </LinearLayout>  
102 </LinearLayout>  
103 </android.support.v7.widget.CardView>  
104  
105 </RelativeLayout>  
106
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     android:layout_width="match_parent" android:
  layout_height="wrap_content"
4     android:paddingLeft="10dp"
5     android:paddingRight="10dp"
6     android:paddingBottom="5dp"
7     android:layout_gravity="center">
8
9     <LinearLayout
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:layout_centerVertical="true"
13        android:orientation="vertical"
14        android:layout_marginRight="35dp">
15        <TextView
16            android:id="@+id/textView_kontroler_tlname"
17            android:layout_width="wrap_content"
18            android:layout_height="wrap_content"
19            android:textAppearance="@style/TextAppearance.
  AppCompatActivity.Medium"
20            android:text="TLname" />
21
22        <TextView
23            android:id="@+id/textView_kontroler_tlloc"
24            android:layout_width="wrap_content"
25            android:layout_height="wrap_content"
26            android:text="TLloc" />
27    </LinearLayout>
28
29    <RadioButton
30        android:layout_width="wrap_content"
31        android:layout_height="wrap_content"
32        android:layout_centerVertical="true"
33        android:layout_alignParentEnd="true"
34        android:id="@+id/radio_kontroler_tlselected"/>
35
36 </RelativeLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context="aplikasiku.navwithdirectionlib.
  HelloFirebaseDB">
9
10    <EditText
11        android:id="@+id/editTextDBname"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:ems="10"
15        android:inputType="textPersonName"
16        android:text="Name"
17        android:layout_marginTop="25dp"/>
18
19    <EditText
20        android:id="@+id/editTextDBstts"
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:ems="10"
24        android:inputType="textPersonName"
25        android:text="Satus"
26        android:layout_marginTop="15dp"/>
27
28    <Button
29        android:id="@+id/buttonInsert"
30        android:layout_width="match_parent"
31        android:layout_height="wrap_content"
32        android:layout_marginTop="45dp"
33        android:text="Insert" />
34
35    <ProgressBar
36        android:id="@+id/create_progress"
37        style="?android:attr/progressBarStyleLarge"
38        android:layout_width="wrap_content"
39        android:layout_height="wrap_content"
40        android:layout_marginTop="15dp"
41        android:layout_gravity="center"
42        android:visibility="invisible" />
43
44    <TextView
45        android:id="@+id/textView2"
46        android:layout_width="match_parent"
47        android:layout_height="wrap_content"
48        android:text="TextView" />
49
50    <EditText
51        android:id="@+id/editTextNameToUpdate"
52        android:layout_width="match_parent"
53        android:layout_height="wrap_content"
```

```
54         android:layout_marginTop="15dp"
55         android:ems="10"
56         android:inputType="textPersonName"
57         android:text="Name to Update" />
58
59     <EditText
60         android:id="@+id/editTextValueUpdate"
61         android:layout_width="match_parent"
62         android:layout_height="wrap_content"
63         android:ems="10"
64         android:layout_marginTop="5dp"
65         android:inputType="textPersonName"
66         android:text="Name Value Update" />
67
68     <Button
69         android:id="@+id/buttonUpdate"
70         android:layout_width="match_parent"
71         android:layout_height="wrap_content"
72         android:layout_marginTop="10dp"
73         android:text="Update" />
74
75 </LinearLayout>
76
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk
  /res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     tools:context="aplikasiku.navwithdirectionlib.
  InserterHospitalActivity">
8
9     <ScrollView
10         android:layout_width="match_parent"
11         android:layout_height="match_parent">
12         <LinearLayout
13             android:layout_width="match_parent"
14             android:layout_height="wrap_content"
15             android:orientation="vertical">
16
17             <!-- dummy item focusable first -->
18             <LinearLayout
19                 android:focusable="true"
20                 android:focusableInTouchMode="true"
21                 android:layout_width="0px"
22                 android:layout_height="0px"/>
23
24             <android.support.design.widget.TextInputLayout
25                 android:layout_width="match_parent"
26                 android:layout_height="wrap_content"
27                 android:layout_marginTop="5dp">
28                 <EditText
29                     android:id="@+id/editTextRSname"
30                     android:layout_width="match_parent"
31                     android:layout_height="wrap_content"
32                     android:hint="RS Name"
33                     android:maxLines="1"
34                     android:nextFocusUp="@id/editTextRSname"
35                     android:nextFocusLeft="@id/
  editTextRSname" />
36                 </android.support.design.widget.TextInputLayout
37             >
38
39             <android.support.design.widget.TextInputLayout
40                 android:layout_width="match_parent"
41                 android:layout_height="wrap_content"
42                 android:layout_marginTop="5dp">
43                 <EditText
44                     android:id="@+id/editTextRSloc"
45                     android:layout_width="match_parent"
46                     android:layout_height="wrap_content"
47                     android:hint="RS Loc"
48                     android:maxLines="1" />
49                 </android.support.design.widget.TextInputLayout
50             >
```

```
50     <android.support.design.widget.TextInputLayout
51         android:layout_width="match_parent"
52         android:layout_height="wrap_content"
53         android:layout_marginTop="5dp">
54         <EditText
55             android:id="@+id/editTextRSlat"
56             android:layout_width="match_parent"
57             android:layout_height="wrap_content"
58             android:hint="RS Lat"
59             android:maxLines="1" />
60     </android.support.design.widget.
TextInputLayout>
61
62     <android.support.design.widget.TextInputLayout
63         android:layout_width="match_parent"
64         android:layout_height="wrap_content"
65         android:layout_marginTop="5dp">
66         <EditText
67             android:id="@+id/editTextRSlong"
68             android:layout_width="match_parent"
69             android:layout_height="wrap_content"
70             android:hint="RS Long"
71             android:maxLines="1" />
72     </android.support.design.widget.
TextInputLayout>
73
74     <Button
75         android:layout_width="match_parent"
76         android:layout_height="wrap_content"
77         android:id="@+id/buttonInsertRSgetGPS"
78         android:layout_marginTop="45dp"
79         android:text="Loc by GPS"/>
80
81     <Button
82         android:layout_width="match_parent"
83         android:layout_height="wrap_content"
84         android:id="@+id/buttonInsertRSok"
85         android:text="Insert RS"/>
86     </LinearLayout>
87 </ScrollView>
88 </LinearLayout>
89
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:layout_margin="20dp">
7
8     <LinearLayout
9         android:layout_width="match_parent"
10        android:layout_height="match_parent"
11        android:orientation="vertical">
12
13        <TextView
14            android:id="@+id/textview"
15            android:layout_width="match_parent"
16            android:layout_height="wrap_content"
17            android:layout_marginTop="1dp"
18            android:gravity="center"
19            android:text="INPUT DATA RS"
20            android:textSize="20dp"
21            android:textStyle="bold" />
22
23        <ImageView
24            android:id="@+id/ShowImageView"
25            android:layout_width="match_parent"
26            android:layout_height="160dp"
27            android:layout_below="@+id/textview"
28            android:layout_marginTop="20dp" />
29
30        <LinearLayout
31            android:layout_width="match_parent"
32            android:layout_height="wrap_content"
33            android:orientation="vertical">
34
35            <android.support.design.widget.TextInputLayout
36                android:layout_width="match_parent"
37                android:layout_height="match_parent"
38                android:layout_marginTop="5dp">
39
40                <EditText
41                    android:id="@+id/editTextRSname"
42                    android:layout_width="match_parent"
43                    android:layout_height="wrap_content"
44                    android:hint="RS Name"
45                    android:maxLines="1"
46                    android:nextFocusLeft="@id/
editTextRSname"
47                    android:nextFocusUp="@id/editTextRSname"
48                />
49            </android.support.design.widget.TextInputLayout
50            >
51            <android.support.design.widget.TextInputLayout
52                android:layout_width="match_parent"
```

```
52         android:layout_height="match_parent"
53         android:layout_marginTop="5dp">
54
55         <EditText
56             android:id="@+id/editTextRSloc"
57             android:layout_width="match_parent"
58             android:layout_height="wrap_content"
59             android:hint="RS Loc"
60             android:maxLines="1" />
61     </android.support.design.widget.
TextInputLayout>
62
63     <android.support.design.widget.TextInputLayout
64         android:layout_width="match_parent"
65         android:layout_height="match_parent"
66         android:layout_marginTop="5dp">
67
68         <EditText
69             android:id="@+id/editTextRSlat"
70             android:layout_width="match_parent"
71             android:layout_height="wrap_content"
72             android:hint="RS Lat"
73             android:maxLines="1" />
74     </android.support.design.widget.
TextInputLayout>
75
76     <android.support.design.widget.TextInputLayout
77         android:layout_width="match_parent"
78         android:layout_height="match_parent"
79         android:layout_marginTop="5dp">
80
81         <EditText
82             android:id="@+id/editTextRSlong"
83             android:layout_width="match_parent"
84             android:layout_height="wrap_content"
85             android:hint="RS Long"
86             android:maxLines="1" />
87     </android.support.design.widget.
TextInputLayout>
88 </LinearLayout>
89
90     <Button
91         android:id="@+id/ButtonChooseImage"
92         android:layout_width="match_parent"
93         android:layout_height="wrap_content"
94         android:layout_below="@+id/ShowImageView"
95         android:text="Choose Image" />
96
97     <Button
98         android:id="@+id/ButtonUploadImage"
99         android:layout_width="match_parent"
100        android:layout_height="wrap_content"
101        android:layout_alignParentBottom="true"
102        android:layout_alignParentStart="true"
103        android:text="Simpan Data RS" />
```

```
104  
105     </LinearLayout>  
106  
107 </RelativeLayout>  
108
```