

LAMPIRAN

IMPLEMENTASI ANTARMUKA

1. Implementasi Antarmuka

Halaman Utama



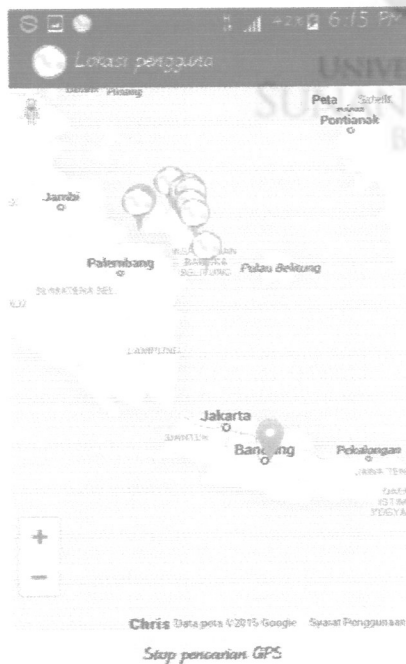
3. Implementasi Antarmuka

Menu Daftar Objek Wisata



2. Implementasi Antarmuka

Menu Lokasi Pengguna



4. Implementasi Antarmuka Peta

Objek Wisata



5. Implementasi Antarmuka Rute



Route :

A Jalan A.H. Nasution No.324, Panyileukan, Kota Bandung, Jawa Barat 40614, Indonesia

791 km - sekitar 1 hari 13 jam

1. Ke arah tenggara di **Jl. A.H. Nasution/Jl. Raya Cipadung** menuju **Bunderan Cibiru** 0,2 km
2. Di **Bunderan Cibiru**, ambil jalan keluar ke-3 menuju **Jl. Nasional III/Jl. Soekarno-Hatta** 8,6 km
3. Belok kiri ke **Jl. Terusan Kiaracondong** 1,0 km
4. Belok kanan menuju **Jl. Terusan Buah Batu** 0,1 km
5. Belok kiri ke **Jl. Terusan Buah Batu** 0,8 km
6. Belok kanan ke **Jalur Keluar Masuk Tol Buah Batu** 0,6 km
Jalan tol sebagian
7. Tetap di kanan di pertigaan dan bergabung ke **Jl. Tol Padaleunyi** 21,2 km
Jalan tol
8. Terus ke **Jl. Tol Cipularang** 55,6 km
Jalan tol

7. Implementasi Antarmuka

Menu Tentang



Aplikasi ini berguna untuk membantu dalam perjalanan yang
lebih baik. Aplikasi ini dapat diunduh di Google Play Store.
Created by: Muhammad Nadek Chama
ID: 123456789
Email: nadekchama@gmail.com



6. Implementasi Antarmuka

Menu Tutorial



1. Buka aplikasi: Nadek: Bangkit Tur
2. Aplikasi memiliki menu yang terdapat, terdapat dalam menu yang GPS, atau menggunakan GPS, kemudian akan muncul peta yang akan di gunakan sebagai
3. Setelah menggunakan aplikasi, akan muncul data mengenai lokasi yang akan di gunakan, kemudian akan muncul daftar lokasi yang akan di gunakan di aplikasi Bangkit.
4. Setelah itu, jika ingin menggunakan aplikasi, kemudian akan muncul data yang akan di gunakan, kemudian akan muncul data yang akan di gunakan, kemudian akan muncul data yang akan di gunakan, kemudian akan muncul data yang akan di gunakan.

UIN
UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI
BANDUNG



MainActivity.java

```
package com.taq.nearestbangkatour;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.taq.nearestbangkatour.R;
import com.taq.nearestbangkatour.Helper.DatabaseHelper;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        DatabaseHelper helper = new DatabaseHelper(this);
        helper.insertOfficeData();
        //button1 my location
        Button btnMenu1 =
(Button)findViewById(R.id.button1);
        btnMenu1.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(),
        Mylocation.class);
        startActivity(intent);
    }
});
        //button2 Nearest Police office
        Button btnMenu2 = (Button)findViewById(R.id.button2);
        btnMenu2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
                BangkaTourList.class);
                startActivity(intent);
            }
        });
        //button3 Nearest Police office
        Button btnMenu3 = (Button)findViewById(R.id.button3);
        btnMenu3.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
                GuidLine.class);
            }
        });
    }
}
```

```

startActivity(intent);
    }
    });
//button4 Nearest Police office
    Button btnMenu4 = (Button)findViewById(R.id.button4);
    btnMenu4.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getApplicationContext(),
            About.class);
            startActivity(intent);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it
    is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public void onDestroy() {
    finish();
    super.onDestroy();
}

@Override
public void onBackPressed() {
    stopService(new Intent(this, GPSService.class));
    Toast.makeText(getApplicationContext(),"Thanks for using
application!!",Toast.LENGTH_SHORT).show();
    finish();
    return;
}
}

```

Mylocation.java

```

package com.taq.nearestbangkatour;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DialogFragment;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.location.LocationManager;
import android.os.Bundle;

```

```

import android.provider.Settings;
import android.text.format.Time;
import android.view.Menu;
import android.view.View;
import android.webkit.WebView;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.taq.nearestbangkatour.R;
import com.taq.nearestbangkatour.Helper.DatabaseHelper;

public class Mylocation extends Activity {
    private TextView latitudeField;
    private TextView longitudeField;
    private LocationManager locationManager;
    private Context mContext;
    private Intent serviceIntent;

    private WebView mapView = null;

    //flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;

    /**Intent Receiver**/
    private BroadcastReceiver receiver = new BroadcastReceiver()
    {
        @Override
        public void onReceive(Context context, Intent intent) {
            Bundle bundle = intent.getExtras();
            if (bundle != null) {
                String lat = bundle.getString("lat");
                String lng = bundle.getString("long");

                latitudeField = (TextView) findViewById(R.id.TextView02);
                longitudeField = (TextView) findViewById(R.id.TextView04);

                latitudeField.setText(lat);
                longitudeField.setText(lng);
                Time time = new Time();
                time.setToNow();

                if (mapView != null){
                    mapView.loadUrl("javascript:updateLocation(\"+lat+\",\"+lng+\", \"\
                    \""+time.format("%d-%m-%Y %H:%M:%S")+\"\\\" \");
                }

                CharSequence text = "GPS Location Updated";
            }
        }
    }
}

```

```

int duration = Toast.LENGTH_SHORT;

DatabaseHelper db = new DatabaseHelper(context);
db.updateMyLocation(-1, lat+", "+lng);
Toast toast = Toast.makeText(context, text, duration);
toast.show();
}
}

/** Called when the activity is first created. */

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mylocation);
    mContext = this;

    serviceIntent = new Intent(this, GPSService.class);

    latitudeField = (TextView) findViewById(R.id.TextView02);
    longitudeField = (TextView) findViewById(R.id.TextView04);

    mapView = (WebView) findViewById(R.id.mapView);
    mapView.getSettings().setJavaScriptEnabled(true);

    mapView.loadUrl("file:///android_asset/maps.html");
    locationManager = (LocationManager) mContext
        .getSystemService(LOCATION_SERVICE);

    // getting GPS status
    isGPSEnabled = locationManager

.isProviderEnabled(LocationManager.GPS_PROVIDER);

    // getting network status
    isNetworkEnabled = locationManager

.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

    if (!isGPSEnabled) {
        DialogFragment confirmGPS = new GPSSettingsDialogFragment();
        confirmGPS.show(getFragmentManager(), "gpsConfirm1");
    } else {
        // if GPS Enabled get lat/long using GPS Services
        if (isGPSEnabled) {
            latitudeField.setText("Location not available");
            longitudeField.setText("Location not available");
        }

        //here start service
        startService(serviceIntent);
    } else {

```

```

DialogFragment confirmGPS = new GPSSettingsDialogFragment();
confirmGPS.show(getFragmentManager(), "gpsConfirm2");
    }
}

Button gpsStopBtn = (Button) findViewById(R.id.stopService);
gpsStopBtn.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {
        //here stop service
        stopService(serviceIntent);
    }

});
}

/* Request updates at startup */
@Override
protected void onResume() {
    super.onResume();
    registerReceiver(receiver, new
IntentFilter(GPSService.COORDSNOTIFICATION));
}

/* Remove the locationlistener updates when Activity
is paused */
@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(receiver);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the
action bar if it is present.
    getMenuInflater().inflate(R.menu.main,
menu);
    return true;
}

/**
 * Function to show settings alert dialog
 * */
public class GPSSettingsDialogFragment extends
DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the Builder class for convenient dialog
constructioAlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
builder.setMessage("GPS is not enabled. Do you want to go to
settings menu?")

```

```

.setPositiveButton("Settings", new
DialogInterface.OnClickListener() {
    public void
onClick(DialogInterface dialog,int which) {
        Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);

mContext.startActivity(intent);
    }

})

.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    } });
// Create the AlertDialog object and return it
return builder.create();
    }

}

```

BangkaTourList.java

```

package com.taq.nearestbangkatour;

import java.util.ArrayList;
import java.util.List;

import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;

import com.taq.nearestbangkatour.CustomListView;
import com.taq.nearestbangkatour.R;
import com.taq.nearestbangkatour.Helper.DatabaseHelper;
import com.taq.nearestbangkatour.Model.BangkaTour;

public class BangkaTourList extends ListActivity {

    private ArrayList<String> listOffice = new
ArrayList<String>();
    private DatabaseHelper helper;
    private List<BangkaTour> listPO, orderedList;
    private ProgressDialog progressDialog;
    private ArrayList<Integer> imgId = new

```

```

ArrayList<Integer>();
    private Integer[] theImage = new Integer[15];

    //ListView myList;

    Integer[] imageId = { //ganti gambar sesuai urutan
R.drawable.s1,R.drawable.s2,R.drawable.s3,R.drawable.s4,R.dr
awable.s5,R.drawable.s6,R.drawable.s7,R.drawable.s8,R.drawab
le.s9,R.drawable.s10,R.drawable.s11,R.drawable.s12,R.drawabl
e.s13,R.drawable.s14,R.drawable.s15
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        new LoadViewTask().execute();
        //myList = (ListView) findViewById(R.id.thelist);
    }

    @Override
    protected void onItemClick(ListView l, View v,
int position, long id) {
        Intent intent = new Intent(getBaseContext(),
NearestBT.class);
        intent.putExtra("officeID", orderedList.get((int)
l.getItemIdAtPosition(position)).getID());
        startActivity(intent);
    }

    private class LoadViewTask extends AsyncTask<Void,
Integer, Void>
    {
        //Before running code in the separate thread
        @Override
        protected void onPreExecute()
        {
            //Create a new progress dialog
            progressDialog =
ProgressDialog.show(BangkaTourList.this,"Loading...",
"Loading List Near Place, please wait...", false, false);
            progressDialog.show();
        }

        //The code to be executed in a background thread.
        @Override
        protected Void doInBackground(Void... params)
        {
            /* This is just a code that delays the thread execution 4
            times,
            *during 850 milliseconds and updates the current progress.
            This
            *is where the code that is going to be executed on a
            background
            * thread must be placed.

```

```

*/
try
{
    //Get the current thread's token
    synchronized (this)
    {
        //Initialize an integer (that will act as a counter) to zero
        int counter = 0;
        //While the counter is smaller than four
        while(counter <= 4)
        {
            //Wait 850 milliseconds
            this.wait(850);
            //Increment the counter
            counter++;
            //Set the current progress.
            //This value is going to be passed to the onProgressUpdate()
            //method.

            publishProgress(counter*25);
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        return null;
    }

    //Update the progress
    @Override
    protected void onProgressUpdate(Integer... values)
    {
        //set the current progress of the progress dialog
        progressDialog.setProgress(values[0]);
    }

    //after executing the code in the thread
    @Override
    protected void onPostExecute(Void result)
    {
        //close the progress dialog
        progressDialog.dismiss();
        //initialize the View
        setContentView(R.layout.activity_bangka_tour_list);

        helper = new DatabaseHelper(getBaseContext());

        listPO = helper.getAllTour();
        Greedy.context = getBaseContext();
        orderedList = Greedy.createGreedy(listPO);
    }
}

```

```

        for(BangkaTour po : orderedList){
            Integer imgId1 = po.getID();
            String nama = po.getName();
            String distance = po.getDistance();
            String duration = po.getDuration();
            try {
                imgId.add(imgId1);
            } catch (Exception e) {
                //Log.e("myDebug", e.toString());
            }
            listOffice.add(nama + "\nJarak: " + distance + "\nWaktu
Tempuh: " + duration);
        }
        try {
            for(int i = 0; i < imageId.length; i++) {
                theImage[i] = imageId[imgId.get(i)];
            }
        } catch (Exception e){
            //Log.e("myDebug", e.toString());
        }

        CustomListView adapter = new
        CustomListView(BangkaTourList.this, listOffice, theImage);

        setListAdapter(adapter);
        getListView().setTextFilterEnabled(true);

    }
}
}

```

CustomListView.java

```
package com.taq.nearestbangkatour;

import java.util.ArrayList;
import com.taq.nearestbangkatour.R;
import android.app.Activity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class CustomListView extends ArrayAdapter<String> {
    //Declarasi
    private final Activity context;
    private ArrayList<String> web = new
ArrayList<String>();
    private final Integer[] imageId;

    public CustomListView(Activity context,
ArrayList<String> web, Integer[] imageId) {
        super(context, R.layout.list_single_data, web);
        this.context = context;
        this.web = web;
        this.imageId = imageId;
    }

    @Override
    public View getView(int position, View view, ViewGroup
parent) {
        LayoutInflater inflater = context.getLayoutInflater();
        //Load Custom Layout untuk list
        View rowView= inflater.inflate(R.layout.list_single_data,
null, true);
        //Declarasi komponen
        TextView txtTitle = (TextView)
rowView.findViewById(R.id.txtList);
        ImageView imageView = (ImageView)
rowView.findViewById(R.id.imgIcon);

        //Set Parameter Value
        txtTitle.setText(web.get(position));
        imageView.setImageResource(imageId[position]);
        return rowView;
    }
}
```

GPSService.java

```
package com.taq.nearestbangkatour;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import com.taq.nearestbangkatour.R;
import android.app.Notification;
import android.app.Notification.Builder;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Binder;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.IBinder;
import android.os.Looper;
import android.os.Message;
import android.os.Process;
import android.util.Log;
import android.widget.Toast;

public class GPSService extends Service {
    private Looper mServiceLooper;
    private ServiceHandler mServiceHandler;
    private Location location;
    private LocationManager locationManager;

    public static final String COORDSNOTIFICATION =
        "com.taq.nearestbangkatour.service.receiver";

    // Send the notification.
    // Sets an ID for the notification
    int mNotificationId = 001;
    // Gets an instance of the NotificationManager service
    NotificationManager mNotifyMgr;

    // The minimum distance to change Updates in meters
    private static final long
    MIN_DISTANCE_CHANGE_FOR_UPDATES = 1; // 5 meters

    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000
    * 60 * 1; // 1 minute
```

```

        // Handler that receives messages from the thread
        private final class ServiceHandler extends Handler
        implements LocationListener {

            public ServiceHandler(Looper looper) {
                super(looper);
            }
            @Override
            public void handleMessage(Message msg) {
                // if GPS Enabled get lat/long using GPS Services
                if (msg.arg1==0){

locationManager.removeUpdates((LocationListener) this);
                    }else {
                        if (location == null) {
locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
locationManager.requestLocationUpdates(
LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                            Log.d("GPS Enabled", "GPS Enabled");
                            if (locationManager != null) {
                                location = locationManager
                                    .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                                // Initialize the location fields
                                if (location != null) {
onLocationChanged(location);
                                    }
                                }
                            }
                        }
                    }
            @Override
            public void onLocationChanged(Location location)
            {

                double lat = (double) (location.getLatitude());
                double lng = (double) (location.getLongitude());

                try {
                    FileWriter fstream = new
FileWriter(Environment.getExternalStorageDirectory().getPath
()+"coOrds.txt",true);
                    BufferedWriter fbw = new BufferedWriter(fstream);

                    fbw.write(String.valueOf(lng)+" "+String.valueOf(lat)+"0");
                    fbw.newLine();
                }
            }
        }
    }
}

```

```

        fbw.close();
    }
    catch (IOException e) {
        Log.e("Exception", "File write failed: " + e.toString());
    }

    Intent i = new Intent(COORDSNOTIFICATION);
    i.putExtra("lat", String.valueOf(lat));
    i.putExtra("long", String.valueOf(lng));

    sendBroadcast(i);

}
@Override
public void onProviderDisabled(String provider) {
    Log.e("Worker Thread", "Disabled provider " + \
        provider);
}
@Override
public void onProviderEnabled(String provider) {
    Log.e("Worker Thread", "Enabled provider " + \
        provider);
}
@Override
public void onStatusChanged(String provider, int
    status, Bundle extras) {
    // TODO Auto-generated method stub
}

@Override
public void onCreate() {

    mNotifyMgr = (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);

    // Display a notification about us starting.  We put an icon
    in the status bar.
    showNotification();

    // Start up the thread running the service.  Note that we
    create a
    // separate thread because the service normally runs in the
    process's
    // main thread, which we don't want to block.  We also make
    it
    // background priority so CPU-intensive work will not
    disrupt our UI.
    HandlerThread thread = new
    HandlerThread("ServiceStartArguments",
    Process.THREAD_PRIORITY_BACKGROUND);

```

```

        thread.start();
// Get the HandlerThread's Looper and use it for our Handler
        mServiceLooper = thread.getLooper();
        mServiceHandler = new ServiceHandler(mServiceLooper);
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int
startId) { Toast.makeText(this, "GPS service starting",
Toast.LENGTH_SHORT).show();

// For each start request, send a message to start a job and
deliver the
// start ID so we know which request we're stopping when we
finish the job
        Message msg = mServiceHandler.obtainMessage();
        msg.arg1 = 1;
        mServiceHandler.sendMessage(msg)
// If we get killed, after returning from here, restart
        return START_STICKY;
    }

    /**
     * Class for clients to access. Because we know
this service always
     * runs in the same process as its clients, we
don't need to deal with
     * IPC.
     */
    public class LocalBinder extends Binder {
        GPSService getService() {
            return GPSService.this;
        }
    }

    @Override
    public void onDestroy() {

        Message msg = mServiceHandler.obtainMessage();
        msg.arg1 = 0;
        mServiceHandler.sendMessage(msg);

        // Cancel the persistent notification.
        mNotifyMgr.cancel(mNotificationId);

        // Tell the user we stopped.
        Toast.makeText(this, "GPS service ended",
Toast.LENGTH_SHORT).show();
    }

```

```

@Override
public IBinder onBind(Intent intent) {
    return null;
}

/**
 * Show a notification while this service is running.
 */
private void showNotification() {

    // Set the icon, scrolling text and timestamp
    Builder mBuilder =
        new Notification.Builder(this)
        .setSmallIcon(R.drawable.ic_launcher)
        .setContentTitle("GPS Service")
        .setContentText("GPS Service Running");

    Intent resultIntent = new Intent(this, MainActivity.class);

    // Because clicking the notification opens a new
    ("special") activity, there's
    // no need to create an artificial back stack.
    PendingIntent resultPendingIntent =
        PendingIntent.getActivity(
            this,
            0,
            resultIntent,
            PendingIntent.FLAG_UPDATE_CURRENT
        );

    mBuilder.setContentIntent(resultPendingIntent);

    // Builds the notification and issues it.
    mNotifyMgr.notify(mNotificationId,
mBuilder.build());
}
}

```

Route.java

```

package com.taq.nearestbangkatour;

import com.taq.nearestbangkatour.R;
import com.taq.nearestbangkatour.Helper.DatabaseHelper;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.webkit.WebChromeClient;

```

```

import android.webkit.WebView;
import android.webkit.WebViewClient;

public class Route extends Activity {

    private Intent getIntent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_route);

        DatabaseHelper dbHelper = new DatabaseHelper(this);
        getIntent = getIntent();
        final String pos1 = dbHelper.getTour(-
1).getPosition();
        final String pos2 =
dbHelper.getTour(getIntent().getInt("officeID")).getP
osition();

        final WebView webView =
(WebView)findViewById(R.id.WebView1);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setAppCacheEnabled(true);
        webView.getSettings().setDatabaseEnabled(true);
        webView.getSettings().setDomStorageEnabled(true);

        webView.setWebChromeClient(new WebChromeClient());
        webView.getSettings().setGeolocationEnabled(true);
        webView.loadUrl("file:///android_asset/rote.html");

        webView.setWebViewClient(new WebViewClient() {
            public void onPageFinished(WebView view, String
loc) {
                webView.loadUrl("javascript:setLocation(\""+pos1+"\", \""+pos
2+"\"");");
            }
        });

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the
            action bar if it is present.
            getMenuInflater().inflate(R.menu.route, menu);
            return true;
        }
    }
}

```

GuideLine.java

```
package com.taq.nearestbangkatour;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;

import com.taq.nearestbangkatour.R;

public class GuidLine extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_guid_line);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
        action bar if it is present.
        getMenuInflater().inflate(R.menu.guid_line,
        menu);
        return true;
    }
}
```

About.java

```
package com.taq.nearestbangkatour;

import com.taq.nearestbangkatour.R;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class About extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
        action bar if it is present.
        getMenuInflater().inflate(R.menu.about, menu);
        return true;
    }
}
```

NearestBT.java

```
package com.taq.nearestbangkatour;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;

import com.taq.nearestbangkatour.R;
import com.taq.nearestbangkatour.Helper.DatabaseHelper;

public class NearestBT extends Activity {

    private Intent getIntent;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nearest_bt);

        DatabaseHelper dbHelper = new DatabaseHelper(this);
        getIntent = getIntent();
        final String pos1 = dbHelper.getTour(-
1).getPosition();
        final String pos2 =
dbHelper.getTour(getIntent.getExtras().getInt("officeID")).getP
osition();

        final WebView webView =
(WebView)findViewById(R.id.WebView1);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setAppCacheEnabled(true);
        webView.getSettings().setDatabaseEnabled(true);
        webView.getSettings().setDomStorageEnabled(true);
        webView.setWebChromeClient(new WebChromeClient());
        webView.getSettings().setGeolocationEnabled(true);

        webView.loadUrl("file:///android_asset/direct.html");

        webView.setWebViewClient(new WebViewClient() {
            public void onPageFinished(WebView view, String
loc) {

                webView.loadUrl("javascript:setLocation(\""+pos1+"\", \""+pos
2+"\"");");
            }
        })
    }
}
```

```

    });

    Button btnMenu2 = (Button)findViewById(R.id.Route);
    btnMenu2.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getApplicationContext(),
            Route.class);
            intent.putExtra("officeID",
            getIntent().getExtras().getInt("officeID"));
            startActivity(intent);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it
is present.
    getMenuInflater().inflate(R.menu.nearest_bt, menu);
    return true;
}
}

```

SplashScreen.java

```

package com.taq.nearestbangkatour;

import com.taq.nearestbangkatour.R;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class SplashScreen extends Activity {

    private static final int SPLASH_TIME = 3 * 1000; // 3 seconds

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splashscreen);

        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {

                Intent intent = new Intent(SplashScreen.this,
                MainActivity.class);
            }
        }, SPLASH_TIME);
    }
}

```

```

        startActivity(intent);

        SplashScreen.this.finish();

        overridePendingTransition(R.anim.fade_in, R.anim.fade_out);
    }
    }, SPLASH_TIME);

    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {
            } }, SPLASH_TIME);}
    @Override
    public void onBackPressed() {
        this.finish();
        super.onBackPressed();
    }
}

```

DatabaseHelper.java

```

package com.taq.nearestbangkatour.Helper;

import java.util.ArrayList;
import java.util.List;

import com.taq.nearestbangkatour.Model.BangkaTour;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DatabaseHelper extends SQLiteOpenHelper{
    SQLiteDatabase db = getWritableDatabase();
    public static final int DATABASE_VERSION = 1;
    public static final boolean debug = true;
    public static final String LOG_TAG = "DBAdapeter";

    public static final String DB_NAME = "PoliceDB";

    public static final String OFFICE_TBL = "PoliceOffice";
    public static final String ID = "office_id";
    public static final String NAME = "office_name";
    public static final String ADDRESS = "office_address";
    public static final String POSITION = "office_position";
    public static final String PHONE = "office_phone";
    public static final String SERVICE = "office_service";
    public static final String POS_CODE = "office_pos_code";

```

```

public static final String CREATE_OFFICE = "CREATE TABLE IF
NOT EXISTS "+OFFICE_TBL+"("+ID+" INTEGER PRIMARY KEY
autoincrement,"+NAME+" TEXT NOT NULL,"+ADDRESS+"
TEXT,"+POSITION+" TEXT NOT NULL,"+PHONE+" TEXT,"+SERVICE+"
TEXT,"+POS_CODE+" TEXT);";

public static final String[] ALL_TABLES = {OFFICE_TBL};

    public DatabaseHelper(Context context){
        super(context, DB_NAME, null, DATABASE_VERSION);
        onCreate(db);
    }

    @Override
    public void onCreate(SQLiteDatabase database){
    }

    @Override
    public void onUpgrade(SQLiteDatabase database, int
oldVersion, int newVersion) {
        if (debug) Log.w(LOG_TAG, "Upgrading database from
version" + oldVersion + "to" + newVersion + "...");

        for (String table : ALL_TABLES) database.execSQL("DROP TABLE
IF EXISTS " + table);

        onCreate(database);
    }

    //Insert
    public void insertOfficeData(){
        for (String table : ALL_TABLES) db.execSQL("DROP TABLE IF
EXISTS " + table);
        db.execSQL(CREATE_OFFICE);
        List<BangkaTour> bangkaTours = new
ArrayList<BangkaTour>();

        bangkaTours.add(new BangkaTour(-1,"My Location",null,"-
6.9325625,107.7160644", null));
        bangkaTours.add(new BangkaTour(0,"Museum Timah","Jl. apa no
berapa",-2.1159948,106.1205269,"44188"));
        bangkaTours.add(new BangkaTour(1,"Pantai Pasir Padi","Jl.
apa no berapa",-2.116966,106.173151,"44163","0262-
233027","deskripsinya mana"));
        bangkaTours.add(new BangkaTour(2,"Bangka Botanical
Garden","Jl. apa no berapa",-
2.120391,106.169579,"44112","0262-233652","deskripsinya
mana"));
        bangkaTours.add(new BangkaTour(3,"Pantai Matras","Jl. apa no
berapa",-1.794044,106.107778,"44182","0262-
442666","deskripsinya mana"));
        bangkaTours.add(new
BangkaTour(4,"Pantai Parai Tenggara","Jl. apa no berapa",-
1.806118,106.125300,"44183","0262-4441110","deskripsinya

```

```

mana"));
angkaTours.add(new BangkaTour(5,"Pantai Pesona","Jl. apa no
berapa",-1.876125,106.171813,"44161","0262-
542266","deskripsinya mana"));
angkaTours.add(new BangkaTour(6,"Pemandian Air Panas Tirta
Tapta Pemali","Jl. apa no berapa",-
1.860653,106.046561,"44191","0262-444575","deskripsinya
mana"));
angkaTours.add(new BangkaTour(7,"Pantai Romodong","Jl. apa
no berapa",-1.653315,105.763270,"44152","0262-
456110","deskripsinya mana"));
angkaTours.add(new BangkaTour(8,"Giri Sana Menumbing","Jl.
apa no berapa",-2.056933,105.180096,"44153","0262-
455110","deskripsinya mana"));
angkaTours.add(new BangkaTour(9,"Pantai Dan Mercusuar
Tanjung Kalian","Jl. apa no berapa",-
2.084006,105.132517,"44185","0262-466110","deskripsinya
mana"));
angkaTours.add(new BangkaTour(10,"Pantai Tanjung Ular","Jl.
apa no berapa",-2.068517,105.125330,"44188","0262-
421110","deskripsinya mana"));
angkaTours.add(new BangkaTour(11,"Masjid Jamik Dan
Kelenteng Kung Fuk Min","Jl. apa no berapa",-
2.064285,105.156327,"44184","0262-444343","deskripsinya
mana"));
angkaTours.add(new BangkaTour(12,"Wisma Ragam","Jl. apa no
berapa",-2.059289,105.163283,"44162","0262-
233181","deskripsinya mana"));
angkaTours.add(new BangkaTour(13,"Desa Nelayan Kurau","Jl.
apa no berapa",-2.331484,106.234577,"44181","0262-
233739","deskripsinya mana"));
angkaTours.add(new BangkaTour(14,"Benteng Tobolai","Jl. apa
no berapa",-
3.0092513,106.4596237,"44163",null,"deskripsinya mana"));

```

```

createOfficeData(angkaTours);
}

```

```

public void createOfficeData(List<BangkaTour> pos){
    ContentValues values = new ContentValues();

```

```

    for(BangkaTour bangkaTour : pos){
        values.clear();
        values.put(ID, bangkaTour.getID());
        values.put(NAME, bangkaTour.getName());
        values.put(ADDRESS, bangkaTour.getAddress());
        values.put(POSITION,
        bangkaTour.getPosition());
        values.put(POS_CODE, bangkaTour.getPosCode());
        values.put(PHONE, bangkaTour.getPhone());
        values.put(SERVICE, bangkaTour.getService());

```

```

        // insert row
        try{

```

```

        db.insert(OFFICE_TBL, null, values);
    }
    catch(Exception e){
        Log.i(LOG_TAG, "Exception: "+e);
    }
}

}

public void createOfficeData(BangkaTour bangkaTour){
    ContentValues values = new ContentValues();

    values.clear();
    values.put(ID, bangkaTour.getID());
    values.put(NAME, bangkaTour.getName());
    values.put(ADDRESS, bangkaTour.getAddress());
    values.put(POSITION, bangkaTour.getPosition());
    values.put(POS_CODE, bangkaTour.getPosCode());
    values.put(PHONE, bangkaTour.getPhone());
    values.put(SERVICE, bangkaTour.getService());

    // insert row
    try{
        db.insert(OFFICE_TBL, null, values);
    }
    catch(Exception e){
        Log.i(LOG_TAG, "Exception: "+e);
    }
}

public void updateMyLocation(int id, String position){
    ContentValues values = new ContentValues();

    values.clear();
    values.put(POSITION, position);

    db.update(OFFICE_TBL, values, ID+"="+id, null);
}

/**
 * getting all policeOffice
 * */
public List<BangkaTour> getAllTour() {
    String selectQuery = "SELECT * FROM " + OFFICE_TBL
+ " WHERE "+ID+" != -1";
    List<BangkaTour> pos = new ArrayList<BangkaTour>();

    Cursor c = db.rawQuery(selectQuery, null);

    // looping through all rows and adding to list
    if (c.moveToFirst()) {
        do {
            BangkaTour po = new BangkaTour();
            po.setID(c.getInt(c.getColumnIndex(ID)));

```

```

po.setName(c.getString(c.getColumnIndex(NAME)));
po.setAddress(c.getString(c.getColumnIndex(ADDRESS)));
po.setPosition(c.getString(c.getColumnIndex(POSITION)));
po.setPosCode(c.getString(c.getColumnIndex(POS_CODE)));
po.setPhone(c.getString(c.getColumnIndex(PHONE)));
po.setService(c.getString(c.getColumnIndex(SERVICE)));

    // Adding contact to list
    pos.add(po);
    } while (c.moveToNext());
}

return pos;
}

public Cursor getAllTourCursor(){
    String selectQuery = "SELECT * FROM " + OFFICE_TBL;
    Cursor c = db.rawQuery(selectQuery, null);
    return c;
}

/**
 * get single policeOffice
 */
public BangkaTour getTour(int tour_id) {

    String selectQuery = "SELECT * FROM " + OFFICE_TBL
+ " WHERE " + ID + " = " + tour_id;

    Cursor c = db.rawQuery(selectQuery, null);

    if (c != null)
        c.moveToFirst();

    BangkaTour po = new BangkaTour();
    po.setID(c.getInt(c.getColumnIndex(ID)));
    po.setName(c.getString(c.getColumnIndex(NAME)));
    po.setAddress(c.getString(c.getColumnIndex(ADDRESS)));
    po.setPosition(c.getString(c.getColumnIndex(POSITION)));
    po.setPosCode(c.getString(c.getColumnIndex(POS_CODE)));
    po.setPhone(c.getString(c.getColumnIndex(PHONE)));
    po.setService(c.getString(c.getColumnIndex(SERVICE)));

    return po;
}
}

```

DijkstraAlgorithm.java

```
package com.taq.nearestbangkatour;

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.HashSet;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Set;

import com.taq.nearestbangkatour.Model.Edge;
import com.taq.nearestbangkatour.Model.Graph;
import com.taq.nearestbangkatour.Model.Vertex;

public class DijkstraAlgorithm {

    @SuppressWarnings("unused")
    private final List<Vertex> nodes;
    private final List<Edge> edges;
    private Set<Vertex> settledNodes;
    private Set<Vertex> unsettledNodes;
    private Map<Vertex, Vertex> predecessors;
    private Map<Vertex, Integer> distance;

    public DijkstraAlgorithm(Graph graph) {
        // create a copy of the array so that we can operate on
        this array
        this.nodes = new ArrayList<Vertex>(graph.getVertexes());
        this.edges = new ArrayList<Edge>(graph.getEdges());
    }

    public void execute(Vertex source) {
        settledNodes = new HashSet<Vertex>();
        unsettledNodes = new HashSet<Vertex>();
        distance = new HashMap<Vertex, Integer>();
        predecessors = new HashMap<Vertex, Vertex>();
        distance.put(source, 0);
        unsettledNodes.add(source);
        while (unsettledNodes.size() > 0) {
            Vertex node = getMinimum(unsettledNodes);
            settledNodes.add(node);
            unsettledNodes.remove(node);
            findMinimalDistances(node);
        }
    }

    private void findMinimalDistances(Vertex node) {
        List<Vertex> adjacentNodes = getNeighbors(node);
        for (Vertex target : adjacentNodes) {
            if (getShortestDistance(target) >
                getShortestDistance(node)
                    + distance.get(node).get(target)) {
                distance.put(target, distance.get(node).get(target) +
                    getShortestDistance(node));
                predecessors.put(target, node);
                unsettledNodes.add(target);
            }
        }
    }
}
```

```

        + getDistance(node, target)) {
            distance.put(target, getShortestDistance(node)
                + getDistance(node, target));
            predecessors.put(target, node);
            unsettledNodes.add(target);
        }
    }

    private int getDistance(Vertex node, Vertex target) {
        for (Edge edge : edges) {
            if (edge.getSource().equals(node)
                && edge.getDestination().equals(target)) {
                return edge.getWeight();
            }
        }
        throw new RuntimeException("Should not happen");
    }

    private List<Vertex> getNeighbors(Vertex node) {
        List<Vertex> neighbors = new ArrayList<Vertex>();
        for (Edge edge : edges) {
            if (edge.getSource().equals(node)
                && !isSettled(edge.getDestination())) {
                neighbors.add(edge.getDestination());
            }
        }
        return neighbors;
    }

    private Vertex getMinimum(Set<Vertex> vertexes) {
        Vertex minimum = null;
        for (Vertex vertex : vertexes) {
            if (minimum == null) {
                minimum = vertex;
            } else {
                if (getShortestDistance(vertex) <
                    getShortestDistance(minimum)) {
                    minimum = vertex;
                }
            }
        }
        return minimum;
    }

    private boolean isSettled(Vertex vertex) {
        return settledNodes.contains(vertex);
    }

    private int getShortestDistance(Vertex destination) {
        Integer d = distance.get(destination);
        if (d == null) {
            return Integer.MAX_VALUE;
        }
    }

```

```

    } else {
        return d;
    }
}

/*
 * This method returns the path from the source to the
selected target and
 * NULL if no path exists
 */
public LinkedList<Vertex> getPath(Vertex target) {
    LinkedList<Vertex> path = new LinkedList<Vertex>();
    Vertex step = target;
    // check if a path exists
    if (predecessors.get(step) == null) {
        return null;
    }
    path.add(step);
    while (predecessors.get(step) != null) {
        step = predecessors.get(step);
        path.add(step);
    }
    // Put it into the correct order
    Collections.reverse(path);
    return path;
}
}

```

TestDijkstraAlgorithm.java

```

package com.taq.nearestbangkatour;

import static org.junit.Assert.*;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

import org.junit.Test;

import com.taq.nearestbangkatour.Model.Edge;
import com.taq.nearestbangkatour.Model.Graph;
import com.taq.nearestbangkatour.Model.Vertex;

public class TestDijkstraAlgorithm {

    private List<Vertex> nodes;
    private List<Edge> edges;

    @Test
    public void testExcute() {

```

```

nodes = new ArrayList<Vertex>();
edges = new ArrayList<Edge>();
for (int i = 0; i < 11; i++) {
    Vertex location = new Vertex("Node_" + i, "Node_" +
i);
    nodes.add(location);
}

addLane("Edge_0", 0, 1, 85);
addLane("Edge_1", 0, 2, 217);
addLane("Edge_2", 0, 4, 173);
addLane("Edge_3", 2, 6, 186);
addLane("Edge_4", 2, 7, 103);
addLane("Edge_5", 3, 7, 183);
addLane("Edge_6", 5, 8, 250);
addLane("Edge_7", 8, 9, 84);
addLane("Edge_8", 7, 9, 167);
addLane("Edge_9", 4, 9, 502);
addLane("Edge_10", 9, 10, 40);
addLane("Edge_11", 1, 10, 600);

// Lets check from location Loc_1 to Loc_10
Graph graph = new Graph(nodes, edges);
DijkstraAlgorithm dijkstra = new
DijkstraAlgorithm(graph);
dijkstra.execute(nodes.get(0));
LinkedList<Vertex> path =
dijkstra.getPath(nodes.get(10));

assertNotNull(path);
assertTrue(path.size() > 0);

for (Vertex vertex : path) {
    System.out.println(vertex);
}

}

private void addLane(String laneId, int sourceLocNo, int
destLocNo,
    int duration) {
    Edge lane = new Edge(laneId, nodes.get(sourceLocNo),
nodes.get(destLocNo), duration);
    edges.add(lane);
}
}

```

Greedy.java

```

package com.taq.nearestbangkatour;

import java.util.ArrayList;

```

```

import java.util.List;
import java.util.concurrent.ExecutionException;

import org.json.JSONArray;
import org.json.JSONException;
import android.content.Context;

import com.taq.nearestbangkatour.Helper.DatabaseHelper;
import com.taq.nearestbangkatour.Model.BangkaTour;

public class Greedy {

    private static final int param = 2;
    public static Context context = null;

    public static List<BangkaTour>
createGreedy(List<BangkaTour> listPOS) {
        int jumlah = listPOS.size();
        float[][] arrayTarget = new float[param][jumlah];
        float[][] arrayGreedy = new float[param][jumlah];
        int[] indexArray = new int[jumlah];

        arrayTarget = getJarakValue(listPOS);
        // for(int i=0; i<arrayTarget.length; i++)
        arrayTarget[i] = randomizeValue(jumlah, (int)
(Math.random()*100), (int) (Math.random()*10));

        arrayGreedy = greedyValue(arrayTarget);
        indexArray = greedyRank(arrayGreedy);

        return swapList(listPOS, indexArray);
    }

    private static float[][] greedyValue(float[][] arrayTarget){
        float[][] newArray = new
float[param][arrayTarget[0].length];
        for(int i=0; i<arrayTarget.length; i++){
            float minMax = (float) arrayMinMax(arrayTarget[i],true);
            float minMax = (float) arrayMinMax(arrayTarget[i],false);

            for(int j=0; j<arrayTarget[i].length; j++) newArray[i][j] =
((((float) arrayTarget[i][j] - minMax) * ((float) 1 -
(float) 0)) / (maxMax - minMax)) + (float) 0;
        }

        return newArray;
    }

    private static float arrayMinMax(float[] arrayTarget,
boolean status){
        if(status){
            float min = 100;
            for(int i=0; i<arrayTarget.length; i++){
                if(arrayTarget[i] < min) min = arrayTarget[i];
            }
        }
    }
}

```

```

return min;
}
else{
    float max = 0;
    for(int i=0; i<arrayTarget.length; i++){
        if(arrayTarget[i] > max) max = arrayTarget[i];
    }
    return max;
}

}

private static int[] greedyRank(float[][] anArray){
    int[] arrayRank = new int[anArray[0].length];
    for(int i=0; i<arrayRank.length; i++) arrayRank[i] = i;

    for(int i=0; i<anArray[0].length; i++){
        float[] newArray = new float[anArray.length];
        for(int j=0; j<newArray.length; j++){
            newArray[j] = anArray[j][i];
        }
        newArray = sortArray(newArray);
        for(int j=0; j<newArray.length; j++){
            anArray[j][i] = newArray[j];
        }
    }
    for(int i=0; i<anArray[0].length; i++){
        for(int j=0; j<anArray[0].length; j++){
            int k = 0;
            boolean status = false;
            do{
                if(anArray[k][i] < anArray[k][j]){
                    int temp = arrayRank[i];
                    arrayRank[i] = arrayRank[j];
                    arrayRank[j] = temp;
                }
                for(int l=0; l<anArray.length; l++){
                    float temp1 = anArray[l][i];
                    anArray[l][i] = anArray[l][j];
                    anArray[l][j] = temp1;
                }
                status = true;
            }
            else{
                if(anArray[k][i] == anArray[k][j]) k++;
                else status = true;
            }
        }while(!status && k<anArray.length);
    }
    return arrayRank;
}

private static float[] sortArray(float[] arrayValue){
    for(int i=0; i<arrayValue.length; i++)
        for(int j=0; j<arrayValue.length; j++)

```

```

        if(i != j)
            if(arrayValue[i] > arrayValue[j]){
                float temp = arrayValue[i];
                arrayValue[i] = arrayValue[j];
                arrayValue[j] = temp;
            }
        return arrayValue;
    }

    private static List<BangkaTour> swapList(List<BangkaTour>
listPOS, int[] arrayIndex){
        List<BangkaTour> pos = new ArrayList<BangkaTour>();

        for(int i=0; i<arrayIndex.length; i++){
            pos.add(listPOS.get(arrayIndex[i]));
        }
        return pos;
    }

    private static float[][] getJarakValue(List<BangkaTour>
POS){
        float[][] returnArray = new float[param][POS.size()];
        JSONArray aJSON;
        DatabaseHelper dbHelper = new DatabaseHelper(context);
        String request =
            "https://maps.googleapis.com/maps/api/distancematrix/json?";
        request += "origins="+ dbHelper.getTour(-
1).getPosition() +"&destinations=";
        int i = 1;
        for(BangkaTour po : POS){
            request += po.getPosition();
            if(i != POS.size()){
                request += "%7C";
            }i++;
        }
        request += "&mode=driving";
        try {
            aJSON = new ReadURLTask().execute(request).get();
            JSONArray element =
                aJSON.getJSONObject(0).getJSONArray("elements");
            for(int il=0; il<element.length(); il++){
                returnArray[0][il] =
                    element.getJSONObject(il).getJSONObject("distance").getInt("
value");
                returnArray[1][il] =
                    element.getJSONObject(il).getJSONObject("duration").getInt("
value");
                POS.get(il).setDistance(element.getJSONObject(il).getJ
SONObject("distance").getString("text"));
                POS.get(il).setDuration(element.getJSONObject(il).getJ
SONObject("duration").getString("text"));
            }
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ExecutionException e) {
            // TODO Auto-generated catch block

```

```
e.printStackTrace();  
} catch (JSONException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
  
return returnArray;  
}  
@SuppressWarnings("unused")  
private static float[] getSecondValue(){  
    return null;  
}  
}
```



uin

UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI
BANDUNG

**SURAT KEPUTUSAN DEKAN FAKULTAS SAINS DAN TEKNOLOGI
UIN SUNAN GUNUNG DJATI BANDUNG**

NOMOR : Un.05/III.7/PP.00.9/0217/2014

**TENTANG
PEMBIMBING PENYUSUNAN SKRIPSI SARJANA/S1**

DEKAN FAKULTAS SAINS DAN TEKNOLOGI UIN SUNAN GUNUNG DJATI BANDUNG

- : Surat permohonan Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung, tanggal 12 Juni 2014, tentang permohonan Penetapan Pembimbing dalam Penyusunan Skripsi/Tugas Akhir (TA) mahasiswa Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung.
- a. bahwa dalam rangka kelancaran pelaksanaan penyusunan Skripsi/Tugas Akhir mahasiswa Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung, setiap mahasiswa memerlukan bimbingan yang sebaik-baiknya, maka perlu ditetapkan pembimbing Skripsi/Tugas Akhir;
- b. bahwa saudara-saudara yang namanya tercantum dalam surat keputusan ini dipandang mampu dan memenuhi syarat untuk menjadi pembimbing penyusunan Skripsi/Tugas Akhir mahasiswa Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung.
1. Undang-undang Nomor 20 Tahun 2003, tentang Sistem Pendidikan Nasional;
 2. Undang-undang Nomor 12 Tahun 2012, tentang Pendidikan Tinggi;
 3. Peraturan Pemerintah Nomor 66 Tahun 2010 tentang Pengelolaan dan Penyelenggaraan Pendidikan;
 4. Peraturan Presiden Nomor 57 Tahun 2005, tentang perubahan IAIN menjadi UIN Sunan Gunung Djati Bandung;
 5. Peraturan Menteri Agama RI Nomor 7 Tahun 2013 tentang Organisasi dan Tata Kerja UIN Sunan Gunung Djati Bandung;
 6. Keputusan Menteri Agama RI Nomor 39 Tahun 2010 tentang Statuta UIN Sunan Gunung Djati Bandung;
 7. Surat Keputusan Rektor UIN Sunan Gunung Djati Bandung, Nomor Un.05/A/Kp.07.6/037.A/2010, tentang pengangkatan Dekan Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung;
 8. Keputusan Rektor UIN Sunan Gunung Djati Bandung Nomor Un.05/L.1/PP.00.9/143/2012 tentang Kurikulum UIN Sunan Gunung Djati Bandung;
 9. Keputusan Rektor UIN Sunan Gunung Djati Bandung, Nomor Un.05/L.1/PP.00.9/161/2013 tentang Pedoman Akademik UIN Sunan Gunung Djati Bandung.

kan : Hasil Seminar/Ujian Proposal Skripsi/Tugas Akhir Jurusan Teknik Informatika UIN Sunan Gunung Djati Bandung, tanggal 22 Mei 2014.

MEMUTUSKAN :

: Terhitung mulai tanggal 17 Juni 2014, mengangkat saudara :

1. Yana Aditia Gerhana, ST., M.Kom. (Pembimbing I)
2. Esa Firmansyah, ST., M.Kom. (Pembimbing II)

dalam penyusunan Skripsi/Tugas Akhir mahasiswa Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung :

N a m a : **Muhammad Nanda Utama**

Nomor Pokok : 1210705092

Jurusan : Teknik Informatika

Judul Skripsi : Pencarian Rute Terdekat Objek Wisata di Pulau Bangka menggunakan Implementasi Algoritma Greedy dan Algoritma Dijkstra menggunakan LBS berbasis Mobile

dengan ketentuan sebagai berikut :

1. Keputusan ini berlaku sejak tanggal ditetapkan sampai dengan lulusnya mahasiswa tersebut dalam ujian munaqasah skripsi;
2. Kepada pembimbing akan diberikan honorarium sesuai dengan ketentuan dan peraturan yang berlaku;
3. Segala sesuatu akan diperbaiki sebagaimana mestinya, apabila dikemudian hari ternyata terdapat kekeliruan dalam penetapan ini.

Ditetapkan di Bandung

tanggal 17 Juni 2014



Dr. M. Subandi, Drs., Ir., MP.

195404241985031004

an :
Rektor UIN Sunan Gunung Djati Bandung;
Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung;
siswa yang bersangkutan.

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN)
SUNAN GUNUNG DJATI BANDUNG
FAKULTAS SAINS DAN TEKNOLOGI

Jalan A.H. Nasution No. 105 Cibiru - Bandung 40614 Telp. 022-7800525 Fax. 022-7803936 website: <http://fst.uinsgd.ac.id>

nomor : Un.05/III.7/PP.00.9/1370/2014
aspek : -
materi : **Permohonan Izin Penelitian**

Bandung, 14 Juli 2014

Kepada Yth.
Kepala Dinas Pariwisata Provinsi Bangka Belitung
Jl. Merdeka No. 4
Pangkal Pinang

Assalaamu'alaikum Wr. Wb.

Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Gunung
Djati Bandung dengan ini menerangkan bahwa :

Nama : **Muhammad Nanda Utama**
NIM : 1210705092
Jurusan : Teknik Informatika
Semester : VIII (delapan)

adalah mahasiswa Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan
Gunung Djati Bandung tahun Akademik 2013/2014, bermaksud mengadakan
penelitian pada Instansi yang Bapak/Ibu/Saudara pimpin.

Adapun kegiatan ini dilaksanakan dalam rangka tugas akhir dengan judul :
**"Pencarian Rute Terpendek Objek Wisata di Pulau Bangka menggunakan
Implementasi Algoritma Greedy dan Algoritma Dijkstra menggunakan LBS
Berbasis Mobile"**, dengan dosen pembimbing :

1. Yana Aditya, ST., M.Kom. (Pembimbing I)
2. Esa Firmansyah, ST., M.Kom. (Pembimbing II)

Untuk itu kami mohon perkenan Bapak/Ibu/Saudara dapat membantu
mahasiswa dalam pelaksanaan penelitian tersebut.

Demikian surat ini kami sampaikan, atas perkenan dan kerjasamanya kami
ucapkan terimakasih.

Wassalaamu'alaikum Wr. Wb.



Dr. H. M. Subandi, Drs., Ir., MP.

695404241985031004

KARTU PEMBINGGAN SKRIPSI
FAKULTAS SAINS DAN TEKNOLOGI

1210705092

Muhammad Nanda UTAMA

Teknik Informatika

Pencarian Rute Terpendek Objek Wisata di Pulau

RAKIGKA Menggunakan Implementasi Algoritma Greedy

dan Algoritma Dijkstra Menggunakan LBS Berbasis Mobile

Tanggal Bimbingan	Hal yang dikonsultasikan	Saran Perbaikan/Penyempurnaan/Tindakan Lanjut	Paraf Pembimbing	
			I	II
26/11/14	Revisi prolog, prolog, Duga, Tugur, di pne prolog	Revisi prolog, Kurnia Duga & Tugur di prolog Latar belakang, Duga, Tugur	me	me
12/12/15	prolog	Duga & Tugur	me	me
16/01/15	prolog I	prolog I - Mr Duga, Duga & Duga	me	me
18/02/15	prolog I	prolog I - Guna guna prolog prolog	me	me
	prolog I	prolog I - Duga & Tugur	me	me
	prolog II	prolog II - Guna guna prolog II Duga / Tugur	me	me
	prolog I	prolog I Latar belakang	me	me
	prolog II	prolog II Guna guna	me	me

Bandung,
Pembimbing II

bimbing I

ADITIA GERHANA, ST., M.kom.

1978 111 72011 011003

Gsa Firmansyah, ST., M.kom.
NIDN: 0426017901

Mengetahui,
Ketua Jurusan Teknik Informatika

H. Cecep Nurul Hani, ST., MT.
NIP. 197804172005011005