

BAB IV

IMPLEMENTASI DAN PEMBAHASAN SISTEM

4.1. Implementasi Sistem

Berdasarkan analisis dan perancangan sistem yang telah dibuat pada bab sebelumnya, maka implementasi merupakan tahap dimana sistem siap pada keadaan sebenarnya.

4.1.1. Class Java dan XML

- Splash Screen

Kelas splash screen ini merupakan halaman awal dari aplikasi yang penulis buat, splash screen ini berjalan hanya 3.000 milisecond dan berisi tentang nama aplikasi yang dikemas dalam dalam sebuah background.

Berikut ini adalah cuplikan program untuk splash screen .xml.

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/splash" >
</LinearLayout>
```

Script diatas adalah potongan program dari file splash.xml, dimana dalam aplikasi ini file xml adalah sebagai layout.

Perintah `android:background="@drawable/splash"` berfungsi untuk memanggil file gambar yang berada pada folder dimana

gambar splash screen tersebut disimpan. Kemudian file splash.xml tersebut akan dipanggil dalam kelas java. Potongan program dari kelas SplashScreen.java dapat dilihat pada potongan script berikut :

```

setContentView(R.layout.splash);
Thread splashThread = new Thread() {
    @Override
    public void run() {
        try {
            int waited = 0;
            while (waited < 3000) {
                sleep(100);
                waited += 100;
            }
        } catch (InterruptedException e) {
            // do nothing
        } finally {
            finish();
            Intent i = new Intent();
                i.setClassName("copyright.firman",
                    "copyright.firman.MainActivity");
startActivity(i);

```

`setContentView(R.layout.splash)` berfungsi untuk memanggil splash.xml yang berada pada layout. Kemudian untuk mengatur waktu tampilan splash screen ini dengan menggunakan perintah `while (waited < 3000)`, maka setelah 3000 milisecond layout ini akan berpindah ke kelas baru, yaitu kelas MainActivity dengan menggunakan perintah `Intent i = new Intent(); i.setClassName("copyright.firman", "copyright.firman.MainActivity");`

- Main Activity

Kelas Main Activity berfungsi sebagai interface dalam aplikasi ini, di dalam kelas ini terdapat form pencarian dan listview. Yang masing – masing mempunyai fungsi yang berbeda – beda. Untuk membuat menu utama ini diperlukan file main.xml dan kelas MainActivity.java, dimana file main.xml nanti akan dijadikan sebagai layout atau interface.

Berikut ini cuplikan program main.xml :

```
<EditText android:id="@+id/cari"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="top"
    android:layout_weight="1"
    android:hint="Cari"/>
<ImageButton android:id="@+id/button_cari"
    android:layout_width="80px"
    android:layout_height="wrap_content"
    android:onClick="myClickHandler"
    android:src="@drawable/search" >
</ImageButton>
</LinearLayout>
<ListView android:id="@+id/listView1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:cacheColorHint="#00000000" />
```

Potongan script kelas MainActivity.java.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    button = (ImageButton)findViewById(R.id.button_cari);
    button.setImageResource(R.drawable.search);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            setSearch();
        }
    });
    Intent intent = null;
    switch (position) {
        case 0:
            intent = new Intent(MainActivity.this,
DaftarToko.class);
```

```
startActivity(intent);
```

Ada dua fungsi layout pada xml yang dipanggil dalam kelas ini, yaitu main.xml dan listview_item_row.xml. main.xml sudah dijelaskan diatas, sedangkan listview_item_row.xml berfungsi untuk memanggil image yang akan ditampilkan pada menu utama. Perintah `listView1.setOnItemClickListener(new OnItemClickListener()` berfungsi untuk melakukan tindakan ketika salah satu listview dipilih. Kemudian dalam script diatas digunakan intent yang diperlukan sebagai pengenalan suatu aktifitas. Selain itu intent juga dapat berpindah kelas pada saat tombol back pada device ditekan maka aplikasi akan kembali ke halaman sebelumnya (bukan keluar aplikasi). Dalam script diatas setelah button cari ditekan maka perintah intent akan mengeksekusi kelas Pencarian.java.

- Web service

Webservice pada sistem ini menggunakan JSON yang digunakan untuk mengambil id masing-masing data dari web server. Kemudian data tersebut akan dikirim kedalam list yang sudah disediakan seperti pada potongan script DaftarToko.java berikut:

```
String link_url = isi + "tokomotor1.php";
JSONParser jParser = new JSONParser();
JSONObject json = jParser.AmbilJson(link_url);
try {
    str_json = json.getJSONArray("info");
```

```

        for (int i = 0; i < str_json.length(); i++) {
            JSONObject ar = str_json.getJSONObject(i);

            String kd_toko = ar.getString("kd_toko");
            String nm_toko = ar.getString("nm_toko");
            String gambar = ar.getString("gambar");
            String alamat = ar.getString("alamat");

```

Kode `String link_url = isi + "tokomotor1.php";` merupakan target koneksi yang menghubungkan `DaftarToko.java` dengan webserver yang berfungsi untuk mengambil data – data yang dibutuhkan sistem.

- Koneksi

`Koneksi.java` berfungsi untuk menentukan target koneksi dari aplikasi mobile ke web server. Berikut ini cuplikan program `Koneksi.java` :

```

public String isi_koneksi()
{
    String isi = "http://10.0.2.2/TokoMotor/";
    //      String isi = "http://labs.tumpeng.co.id/tokomotor/";
    return isi;
}

```

Pada potongan program diatas berisi alamat url yang akan dipanggil. `http://10.0.2.2/TokoMotor/` merupakan alamat yang dipanggil pada server `local/ localhost`, sedangkan `http://labs.tumpeng.co.id/tokomotor/` merupakan alamat web server yang terhubung dengan sistem.

- Menampilkan Peta

Kelas ini berfungsi untuk menangkap lokasi pengguna dan lokasi toko, serta menampilkan rute perjalanan dari lokasi pengguna ke toko.

Potongan script kelas menampilkan_peta.java dapat dilihat pada script berikut.

```

getFragmentManager().findFragmentById(R.id.map))
        .getMap();
    if (map != null) {
        toko = map.addMarker(new MarkerOptions()
            .position(TOKO)
            .title(nm)
            .snippet(alamat)
            .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.marker)));
        toko.showInfoWindow();
        markers.add(toko);
        map.moveCamera(CameraUpdateFactory.newLatLngZoom(TOKO,
15));
    }
    //get location
    getGps();
    double d = distance(latitude, longitude,
Double.parseDouble(lat), Double.parseDouble(lon));
    if(d > 5000){
        tvAlamat.setVisibility(View.GONE);
        tvJarak.setVisibility(View.GONE);
        tvLebih.setVisibility(View.VISIBLE);
    }else{
        LatLng LOKASIKU = new
LatLng(latitude, longitude);
        lokasiku = map.addMarker(new MarkerOptions()
            .position(LOKASIKU)
            .title("lokasi saya")
            .icon(BitmapDescriptorFactory
                .fromResource(R.drawable.people)));
        markers.add(lokasiku);
        new GetDirection().execute();
    }
}
private void getGps() {
    gps = new GPSTracker(memampilkan_peta.this);
    if (gps.canGetLocation()) {
        latitude = gps.getLatitude();
        longitude = gps.getLongitude();
    } else {
        gps.showSettingsAlert();
    }
}
class GetDirection extends AsyncTask<String, String, String>
{
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
}

```

```

        progressDialog = new
ProgressDialog(menampilkan_peta.this);
        progressDialog.setMessage("Loading route. Please
wait...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }
    @protected String doInBackground(String...
args) {
        String urlString =
"http://maps.googleapis.com/maps/api/directions/json?origin="+Double
.toString(latitude)+" "+Double.toString(longitude)+"&destination
="+lat+" "+lon+"&sensor=false";

```

LatLng TOKO = new LatLng berfungsi untuk memanggil nilai latitude dan longitude yang berada pada database. Karena mulai maret 2013 google mengupdate mapnya menjadi versi 2, maka penggunaan google api versi 1 sudah tidak bisa. Untuk itu diperlukan mapfragment pada program diatas.

```

map=(MapFragment)getManager().findFragmentById(R.id.map))
.getMap();

```

Fungsi `map.addMarker(new MarkerOptions())` adalah untuk memberi tanda pada lokasi.

```

map.moveCamera(CameraUpdateFactory.newLatLngZoom(TOKO, 15));

```

berfungsi untuk menggerakkan kamera agar berada pada posisi lokasi yang kita cari berada di tengah layar, dengan zoom factor yang bernilai 15. `gps = new GPSTracker(menampilkan_peta.this);`

berfungsi untuk menangkap nilai latitude dan longitude yang ada pada kelas menampilkan peta. Kemudian untuk menampilkan rute perjalanan digunakan `class GetDirection extends AsyncTask<String, String, String>` yang berfungsi untuk mengambil informasi rute.

<http://maps.googleapis.com/maps/api/directions/json?origin=>

Potongan url pada script diatas berfungsi untuk menghubungkan koneksi ke Google Map.

- Android Manifest.xml

File AndroidManifest.xml berfungsi untuk mendeskripsikan variable global dari paket aplikasi yang kita gunakan. Kode program AndroidManifest.xml dapat dilihat pada potongan script berikut.

```
<uses-permission
android:name="com.vogella.android.locationapi.maps.permission.MAPS
_RECEIVE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSE
RVICES" />
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
  <application
    android:allowBackup="true"
    android:icon="@drawable/bsf"
    android:label="@string/app_name"
    android:theme="@style/MyCustomTheme" >
    <activity
      android:name="copyright.firman.SplashScreen"
      android:label="@string/app_name" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN"
/>
          <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
      <activity android:name="copyright.firman.MainActivity"
></activity>
<activity android:name="copyright.firman.DaftarToko" ></activity>
  <activity
android:name="copyright.firman.DetailAksesServer" ></activity>
```



```

<activity android:name="copyright.firman.menampilkan_peta" >
  </activity>
<activity android:name="copyright.firman.DaftarBarang"></activity>
<activity android:name="copyright.firman.DetailDaftarBarang" >
  </activity>
<meta-data android:name="com.google.android.maps.v2.API_KEY"
android:value="AIzaSyCAxMJwTgMabSi1GxOPLUkzwcwxk5XYA1U" />

<activity android:name="copyright.firman.Pencarian"
android:label="@string/title_activity_pencarian" ></activity>
<activity android:name="copyright.firman.PolyMap"
android:label="@string/title_activity_poly_map" ></activity>
<activity android:name="copyright.firman.DaftarTokoBarang"
android:label="@string/title_activity_daftar_toko_barang" >
  </activity>
</application>
</manifest>

```

Dalam AndroidManifest.xml ini berisi semua pendefinisian hal-hal yang dibutuhkan oleh aplikasi yang dibuat, mulai dari target versi android yang akan digunakan, akses internet, akses GPS, sampai pendefinisian kelas kelas utama yang dipakai dalam aplikasi.

Untuk mengkoneksikan android ke GoogleMap diperlukan API Key seperti pada potongan script berikut :

```

<meta-data android:name="com.google.android.maps.v2.API_KEY"
android:value="AIzaSyCAxMJwTgMabSi1GxOPLUkzwcwxk5XYA1U" />

```

Pada potongan script diatas terdapat Key "AIzaSyCAxMJwTgMabSi1GxOPLUkzwcwxk5XYA1U" yang didapatkan dengan mendaftarkan MD5 FingerPrint pada masing – masing komputer atau laptop di Google ke alamat <http://code.google.com/intl/id-ID/android/maps-api-signup.html>

4.1.2. Program Pendukung Database

Dalam aplikasi ini dibutuhkan web server yang berfungsi untuk mengambil data dari databases yang kemudian akan ditangani oleh script PHP. Berikut beberapa potongan script PHP yang digunakan dalam membangun aplikasi :

- Koneksi Database

```
<?php
$h = "localhost";
$u = "root";
$p = "";
$d = "tokomotordb";
mysql_connect($h,$u,$p);
mysql_select_db($d); ?>
```

Kode program diatas merupakan perintah untuk melakukan koneksi database, perintah "mysql_connect(\$h,\$u,\$p)" digunakan untuk melakukan koneksi ke database yang terdapat pada server "localhost" dengan username "root" dan password "".

Perintah "mysql_select_db(\$d)", berfungsi untuk memilih database yang akan diakses. Didalam perintah ini database yang dipilih adalah "tokomotordb".

- Data Toko

Perintah yang digunakan untuk melakukan pencarian data pada tabel toko, perintah ini yang berfungsi pada Pencarian.java:

```
if (isset($_GET['q'])) {
    $qs = $_GET['q'];
```

```

        $q = mysql_query("select * from toko where nm_toko
like '%$qs%'");
    }

```

Perintah untuk mengambil data dari tabel relasi yang berfungsi untuk menampilkan toko yang yang terhubung dengan data barang :

```

    if (isset($_GET['b'])) {
        $qs = $_GET['b'];
        $q = mysql_query("select * from relasi r join toko t
on t.kd_toko = r.kd_toko where r.kd_barang = '$qs'");
    }

```

- Detail Toko

Perintah untuk mengambil detail toko dan mengambil data latitude, longitude dari database yang kemudian akan dikirim kepada script java. Potongan detail_tokomotor.php dapat dilihat pada script dibawah ini.

```

function tambah_nama_lokasi(){
    $json = $_SERVER['HTTP_JSON'];
    $data = json_decode($json);
    $latitude = $data->latitude;
    $longitude = $data->longitude;
    $nm_toko = $data->nm_toko;
    $alamat= $data->alamat;

    $hasil = database("INSERT INTO toko(latitude, longitude,
nm_toko,                                alamat)                                VALUES
('$latitude','$longitude','$nm_toko','$alamat')");
    if( $hasil ==1 ) echo "Berhasil menambahkan lokasi
$nm_toko";
}
function info_lokasi(){
    $json = $_SERVER['HTTP_JSON'];
    $data = json_decode($json);
    $latitude = $data->latitude;
    $longitude = $data->longitude;

```

```

        database("INSERT INTO toko(latitude, longitude) VALUES
('$latitude','$longitude')");
    }
    function tampil_lokasi(){
        $json = $_SERVER['HTTP_JSON'];
        $data = json_decode($json);
        $dicari = $data->dicari;
        if($dicari != ''){
            $nm_toko = " ";
            $lokasi = database("SELECT latitude, longitude FROM 'toko' WHERE
kd_toko = ".$kd." ");
            $latitude = '0';
            $longitude = '0';
            while($d =mysql_fetch_array($lokasi)){
                $latitude = $d['latitude'];
                $longitude = $d['longitude'];
            }
        }
    }
}

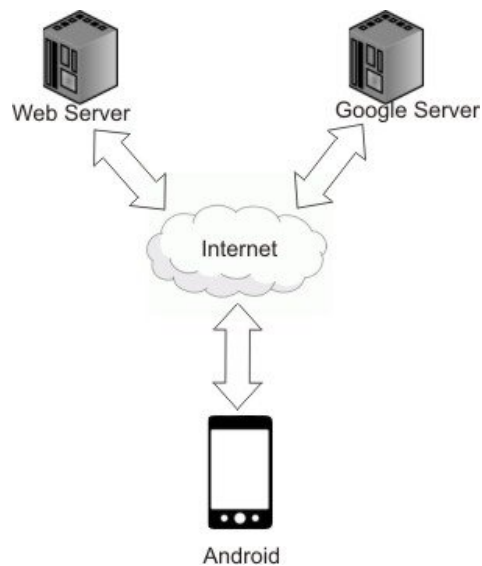
```

4.2. Pembahasan Sistem

Pada aplikasi ini sistem sudah menggunakan web service yang akan memudahkan pembuat aplikasi atau yang sudah diberikan hak akses untuk mengupload isi dari database dari sistem yang dibuat. Sehingga memudahkan untuk menambah, menghapus, dan memperbarui data tanpa harus melakukan pembongkaran aplikasi. Penggunaan web service juga membuat ukuran aplikasi lebih kecil ketika diinstal pada smartphone.

Komunikasi client server bagaimana sistem bekerja secara umum dapat dilihat pada gambar 4.1. Perangkat android mengirim permintaan dan menerima data toko, barang, map. Web server menerima permintaan data toko, barang, map dari perangkat android, menerima permintaan koordinat toko dari

google server. Google server menerima latitude dan longitude posisi pengguna, lokasi toko kemudian mengirim kembali berupa tampilan map.



Gambar 4.1 Komunikasi Client Server

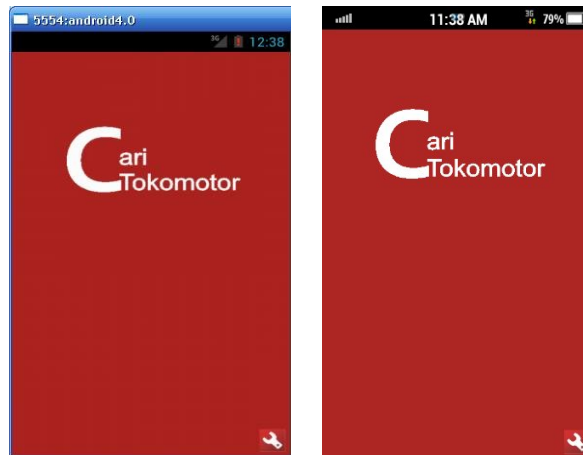
4.2.1. Pengujian Program

Pengujian aplikasi akan dilakukan dengan menggunakan AVD (Android Virtual Device) atau disebut juga emulator dan juga dilakukan pada smartphone Smartfren Andromax E860 dengan sistem operasi android versi 4.0 ICS (Ice Cream Sandwich).

4.2.2. Tampilan Splash Screen

Splash screen adalah menu yang menampilkan background dan nama aplikasi dengan durasi 3000 miliseconds. Ditampilkan pada saat program pertama kali dijalankan sebelum masuk

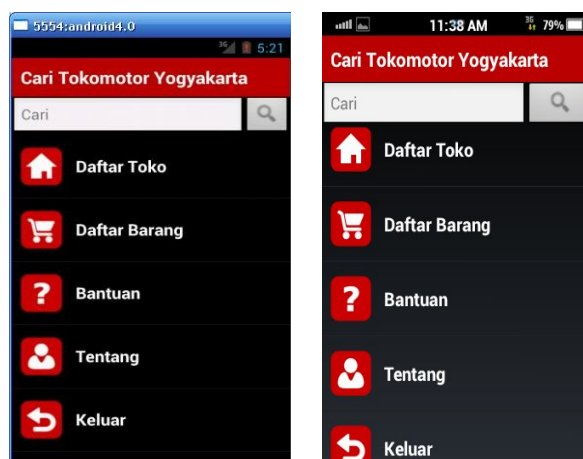
menu utama. Tampilan splashscreen dapat dilihat pada gambar 4.2.



Gambar 4.2 Splashscreen

4.2.3. Tampilan Menu Utama

Tampilan menu utama merupakan tampilan yang berisikan menu pencarian, listview atau daftar list menu seperti daftar toko, daftar barang, bantuan, tentang, dan keluar. Tampilan menu utama dapat dilihat pada gambar 4.3.

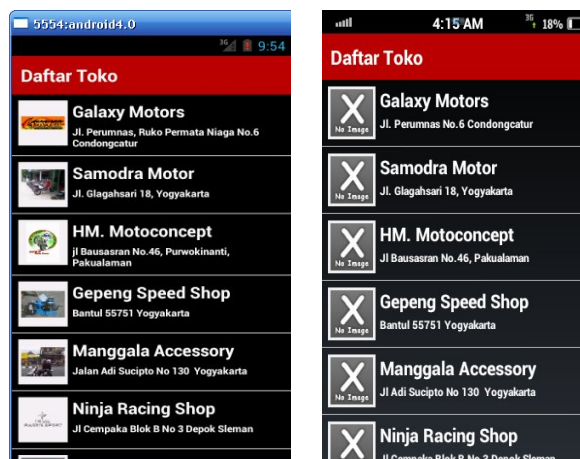


Gambar 4.3 Tampilan Menu Utama

Menu pencarian berfungsi untuk mencari data yang diinginkan pengguna aplikasi dengan memasukan kata kunci yang akan dikirim ke web server. Web server akan ditangani oleh script PHP untuk menerima data, dan selanjutnya akan membangkitkan perintah SQL dalam bentuk query yang memerintahkan untuk mencarinya pada database yang ada pada sistem. Kemudian sistem akan mengeksekusi query dan mengembalikan hasilnya kepada script PHP untuk selanjutnya akan ditampilkan pada aplikasi cari tokomotor dalam bentuk tampilan listview.

4.2.4. Tampilan Daftar Toko

Tampilan daftar toko aksesoris sepeda motor ditampilkan menggunakan listview dengan menampilkan gambar, nama, dan alamat toko. Tampilan daftar toko dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Daftar Toko

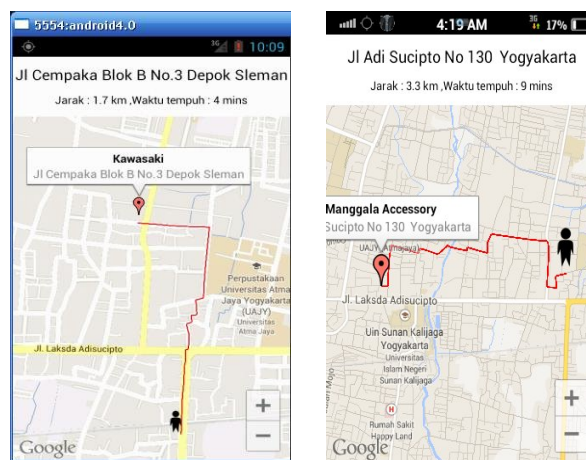
Daftar toko berisi data – data toko yang disimpan pada database yang dapat diisikan melalui web yang sudah dibuat. Selanjutnya data yang diisikan tersebut dikirim ke web server dan pada web server akan ditangani oleh script PHP. Script PHP menerima data, dan selanjutnya akan membangkitkan perintah SQL dalam bentuk query. Kemudian sistem akan mengeksekusi query dan mengembalikan hasilnya kepada script PHP untuk selanjutnya akan ditampilkan pada aplikasi cari tokomotor. Jika pengguna sistem ingin memilih jenis toko tertentu berdasarkan nama toko yang diinginkan, pengguna dapat menggunakan menu pencarian yang ada di menu utama dengan memasukan kata kunci nama toko yang dicari sehingga sistem hanya akan menampilkan daftar toko tertentu sesuai dengan yang diinginkan pengguna sistem.

4.2.5. Tampilan Map

Tampilan map merupakan tampilan yang berlatarkan peta dengan tombol tambah dan tombol kurang yang digunakan untuk memperbesar dan memperkecil tampilan peta serta bar informasi yang memberikan informasi nama, alamat, jarak menuju toko, dan waktu tempuhnya.

Pada tampilan lokasi toko ini akan ditampilkan posisi pengguna sistem dan posisi lokasi toko dengan perwakilan

gambar sebagai posisi pengguna dan lokasi toko sesuai dengan lokasi toko yang dipilih, pengguna cukup menekan tombol temukan lokasi yang terdapat pada detail toko yang dipilih sebelumnya sehingga akan didapat tampilan seperti yang terlihat pada gambar 4.5 berikut:



Gambar 4.5 Tampilan Map

Rute perjalanan pada gambar 4.5 diatas nantinya akan mengikuti kemana pengguna sistem bergerak dan nantinya rute perjalanan baru akan ditampilkan setelah GPS mendapatkan posisi pengguna yang baru. Rute perjalanan didapatkan dari jarak terdekat pengguna sistem ke lokasi toko dengan proses GPS mengirim koordinat pengguna sistem yang akan ditangkap oleh satelit untuk dikembalikan kepada sistem, kemudian sistem akan mencari titik koordinat terdekat dari posisi pengguna yang

dihubungkan ke titik koordinat terdekat berikutnya hingga sampai ke titik koordinat lokasi toko.

4.2.6. Distribusi Program

Distribusi aplikasi Cari Tokomotor dibutuhkan agar pengguna smartphone yang belum mengerti lokasi – lokasi toko aksesoris sepeda motor dapat menggunakan aplikasi ini untuk mencari informasi dan toko yang menjual barang yang diinginkan. Juga dapat memudahkan dalam mencari kebutuhan – kebutuhan aksesoris sepeda motor yang dicari.

Aplikasi ini dapat disebarakan melalui transfer bluetooth antar device android dan juga dapat diajukan untuk dipasarkan melalui playstore.