

# APLIKASI Pencarian Rute dan Site BTS (BASE TRANSCEIVER STATION) Menggunakan Speech to Text pada Perangkat Android

Tiara Anggun Nur Fadhilah<sup>1</sup>, Aisah<sup>2</sup>, Hudiono<sup>3</sup>

Jaringan Telekomunikasi Digital, Teknik Elektro, Politeknik Negeri Malang

## Abstrak

Bisnis telekomunikasi yang saling berkaitan, sangat didasarkan pada pihak sub contractor (subcon)/third party agar terlaksana pengembangan teknologi suatu perangkat BTS (*Base Transceiver Station*) dari sebuah proyek suatu vendor dan operator. Memanfaatkan layanan yang sudah disediakan oleh *smartphone android* yaitu *google maps* beserta *voice* yaitu *speech to text* dapat semakin membantu kinerja dalam suatu tim survei pada pihak *sub contractor* (subcon)/third part.

Hasil pengujian aplikasi dan analisis diperoleh bahwa, pengujian *speech to text* terhadap kebisingan yaitu rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 63 dB sebanyak 79% keberhasilannya, rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 70 dB sebanyak 75% keberhasilannya, rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 75 dB sebanyak 72% keberhasilannya. Pengujian waktu saat membuka *maps* dari hasil *text* ke *maps* didapatkan bahwa kecepatan suara laki-laki lebih cepat yaitu 1 menit 51 detik dan 1 menit 68 detik daripada perempuan. Terakhir, pengujian perbandingan jarak antara aplikasi dengan perhitungan melalui rumus Euclidean Distance diperoleh rata-rata untuk jarak pada aplikasi yang dibuat yaitu 3,65345287 km dan hasil perhitungan dengan rumus Euclidean Distance yaitu 3,66535889 km.

**Kata Kunci :** Aplikasi, Rute, Site, BTS, Speech To Text, Android, Euclidean Distance.

## I. PENDAHULUAN

### 1.1 Latar Belakang

Bisnis telekomunikasi yang terdiri dari operator, pemerintahan, (*government*), vendor, dan *sub contractor* (*subcon*)/third party. *Sub contractor* memiliki 4 tim, yaitu tim survei, tim instalasi, tim integrasi/tim *engineer*, dan tim dokumen. Tim survei berfungsi sebagai tim yang melakukan, mencatat semua informasi dan melakukan dokumentasi berupa foto *sitedan* BTS. Pada saat melakukan survei ke suatu *site* tim survei harus mengetahui *site* tersebut, menggunakan titik *longitude* dan *latitude*, untuk mengetahui *rutedan site*. Titik tersebut akan muncul jika masuk pada *google maps* pada perangkat *smartphone android*. Perangkat *smartphone android*, sangat membantu dalam kegiatan apapun seperti kegiatan tim survei untuk pencarian dan pengambilan data pada *site*.

*Smartphone Android* merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. *Smartphone Android* menyediakan layanan antara lain berupa *voice*, *speech to text*, *maps*, web, google, SMS, telepon, dan masih banyak lagi. Prinsip layanan *speech to text* yaitu menerima gelombang suara manusia lalu mengidentifikasi setiap karakter pada katanya lalu *menconvertnya* menjadi sebuah *text* (Hayatun dan Noviatu, 2016). Manfaat dari layanan ini adalah mengurangi kesalahan memasukkan input pada bidang teks (Supriyanta, Pudji dan Beksti, 2014). Layanan *google maps* dapat membantu tim survei dalam mengetahui *rute* dan *site* BTS. Melalui API *Google Maps*, aplikasi yang

dibuat dapat menampilkan *maps*. Serta API *google maps direction* untuk menampilkan *rute* pada *maps*.

Berdasarkan latar belakang tersebut, maka dibutuhkan suatu aplikasi yang memberi informasi tentang pencarian *rute* dan *site* menggunakan *google maps* serta layanan *google speech to text* yang sudah tersedia di perangkat *smartphone android*. Aplikasi tersebut untuk mempermudah tim survei dalam menemukan *site* BTS. Sistem aplikasi tersebut terdiri dari *insert menu*, *speech menu*, *help menu*. *Insert menu* untuk penambahan *site* baru dengan cara memasukkan *speech* (kode panggilan *site*), *site Id* (nama *site Id*), titik *longitude* dan titik *latitude*. *Speech menu* untuk pengambilan suara *site* yang akan dilanjutkan ke dalam *google maps* dengan cara mengucapkan kode *site* yang sesuai dengan database yang ada, dimana setiap *site* telah diubah kode pemanggilannya. *Help menu* untuk memberitahu tata cara penggunaan aplikasi.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan tersebut maka rumusan masalah sebagai berikut:

1. Bagaimana melakukan pengujian terhadap aplikasi informasi pencarian *rute* dan *site* BTS menggunakan *speech to text* pada perangkat *android* ?
2. Bagaimana perbandingan antara jarak pada aplikasi dengan perhitungan jarak menggunakan rumus teori Euclidean Distance ?

### 1.3 Tujuan Penelitian

Tujuan penelitian adalah berikut :

1. Mendeskripsikan pengujian terhadap aplikasi informasi pencarian *route* dan *site* BTS menggunakan *speech to text* pada perangkat *android*.

## II. TINJAUAN PUSTAKA

### 2.1 Aplikasi

Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti enerapan, penggunaan, dan penambahan data.<sup>[3]</sup>

### 2.2 API(Application Programming Interface)

API (*Application Programming Interface*) atau Antarmuka Pemrograman Aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu.<sup>[4]</sup>

### 2.3 SITE

*Site* adalah tempat atau letak berdirinya sebuah tower BTS dari operator yang di dalamnya terdapat sebuah perangkat dari pihak operator serta vendor.

### 2.4 BTS

BTS adalah bagian dari network element GSM (*Global System Mobile*) yang berhubungan langsung dengan MS (*Mobile Station*). BTS berhubungan dengan MS melalui air-interface dan berhubungan dengan BSC (*Base Station Control*) dengan menggunakan A-bis interface. BTS berfungsi sebagai pengirim dan penerima (transceiver) sinyal komunikasi dari/ke MS serta menghubungkan MS dengan network element lain dalam jaringan GSM (BSC, MSC, SMS, IN, dsb) dengan menggunakan radio interface. Secara hirarki, BTS akan terhubung ke BSC, dalam hal ini sebuah BSC akan mengontrol kerja beberapa BTS yang berada di bawahnya. Karena fungsinya sebagai transceiver, maka bentuk fisik sebuah BTS pada umumnya berupa tower dengan dilengkapi antenna sebagai transceiver, dan perangkatnya. Sebuah BTS dapat mecover area sejauh 35 km (hal ini sesuai dengan nilai maksimum dari Timing Advance (TA)).<sup>[5]</sup>

### 2.5 Speech To Text

Speech To Text adalah Merupakan salah satu tipe dari aplikasi *speech synthesis* digunakan untuk membuat versi suara dari teks mencakup juga *voice-enabled e-mail* dan perangkat suara pada sistem dengan *respons* suara. *Speech to text* juga sering kali digunakan bersamaan dengan program *voice recognition*. *voice recognition* adalah suatu sistem yang dapat mengidentifikasi seseorang melalui suaranya, *voicerecognition* tidak mengidentifikasi siapa yang akan berbicara, tetapi *speech recognition* mengidentifikasi apa yang diucapkan.<sup>[6]</sup>

### 2.6 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem

operasi, *middleware*, dan aplikasi.” Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri.<sup>[7]</sup>

### 2.7 GPS

GPS (Global Positioning System) adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (Departemen of Defense) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit.<sup>[8]</sup>

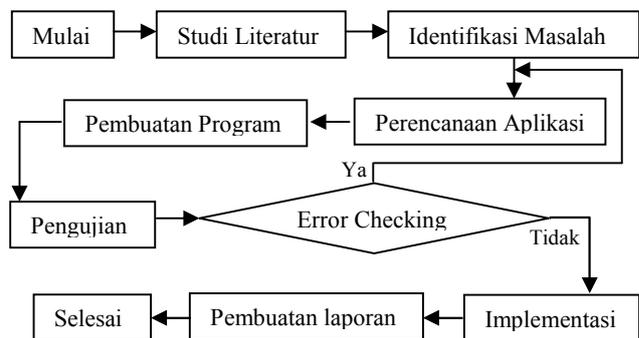
### 2.8 Teorema Euclidean Distance

*Euclidean distance* adalah perhitungan jarak dari 2 buah titik dalam *Euclidean space*. Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tetapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi, dengan rumus dbawah ini.<sup>[9]</sup>

$$\text{Jarak} = 111,319 \times \sqrt{(\text{Lat}_1 - \text{Lat}_2)^2 + (\text{Long}_1 - \text{Long}_2)^2}$$

## III. METODE PENELITIAN

### 3.1 Tahapan Penelitian



Gambar 3.1 Tahapan penelitian

Penjelasan Gambar 3.1 adalah sebagai berikut:

#### 1. Studi literature

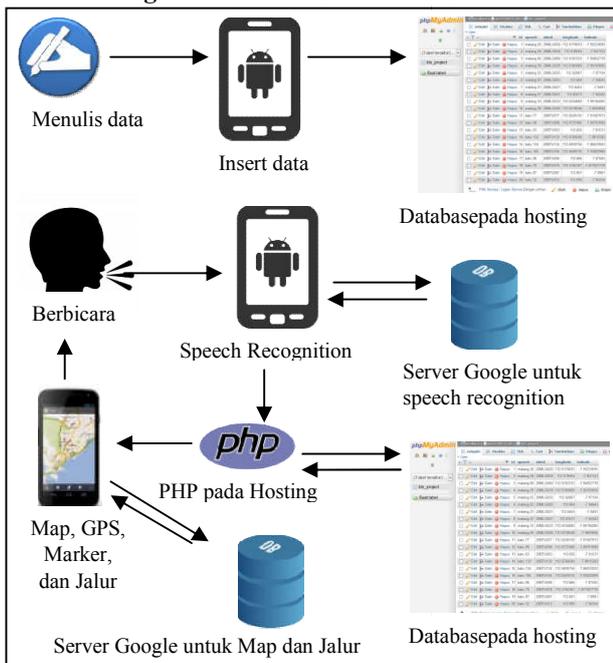
Studi literature ini bertujuan untuk memperoleh data menggunakan fasilitas jaringan internet untuk mencari bahan-bahan yang dapat dijadikan referensi serta buku-buku yang berhubungan dengan pemrograman aplikasi mobile berbasis android serta mewawancarai pihak operator Indosat untuk mengupulkan data mengenai titik koordinat *site* yang dimiliki oleh suatu BTS (*Base Transceiver Station*).

#### 2. Identifikasi Masalah

Berisi mengenai diskripsi masalah kemudian memberi solusi terbaik. Solusi yang didapat yaitu mempermudah pihak sucon khususnya pada tim survei untuk mendapatkan *route* dan *site* BTS. Identifikasi masalah bertujuan untuk memperjelas mengenai aplikasi yang akan dibuat.

3. Perencanaan aplikasi dan pembuatan program  
Bagian ini mengenai software yang berisi kumpulan *longitude* dan *latitude site* beserta detail yang dimiliki suatu BTS. Setelah semua data telah dijadikan dalam bentuk *map* untuk mendapatkan solusi terbaik, maka diimplementasikan dalam system menggunakan bahasa pemrograman *java* agar dikenali oleh pihak sistem operating *android*.
4. Pengujian  
Pengujian yang dimaksudkan untuk menguji aplikasi yang telah dibuat. Apakah telah berjalan dengan baik dan sesuai kenyataan, dengan cara mengetahui respon pihak subcon khususnya tim survei tentang aplikasi tersebut.
5. Pembuatan laporan  
Merupakan tahap terakhir yang berisi hasil laporan serta kesimpulan dari aplikasi yang telah dibuat. Dari sini dapat berisi saran untuk mengembangkan sistem lebih lanjut.

### 3.2 Diagram Alir Pembuatan Sistem

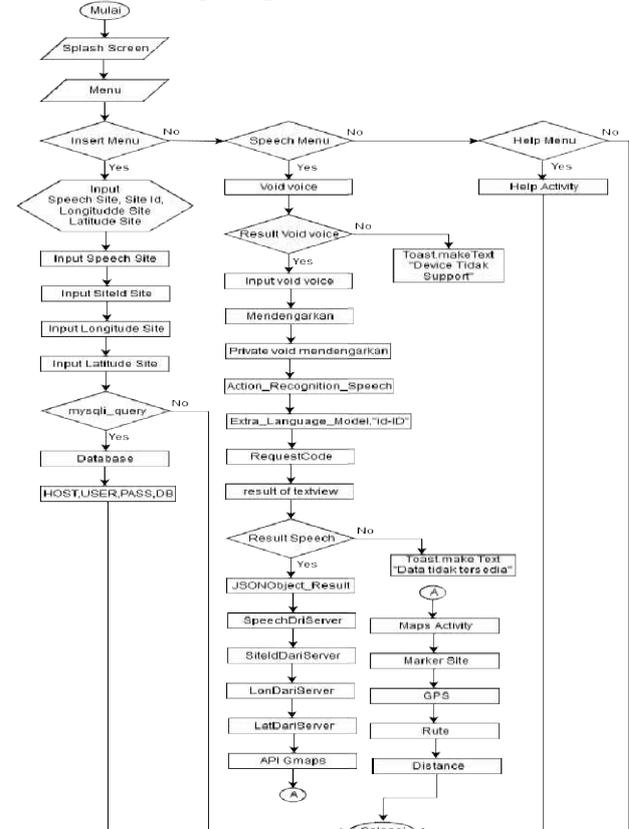


Gambar 3.2 Diagram alir pembuatan sistem penelitian

Gambar 3.2 tentang penjelasan lebih lanjut dari gambar yang sebelumnya yaitu diagram alir pembuatan sistem, dengan melalui *software* Android Studio, maka dibuatlah aplikasi pencarian *route* dan *site* BTS menggunakan *speech to text*. Langkah awal yaitu membuat perintah untuk *insert menu* yang mana data tersebut akan disimpan di data base *hosting* pribadi. Setelah *insert data*, maka membuat perintah untuk *speech recognition* untuk memanggil data yang telah di masukkan. Terakhir, perintah untuk menampilkan peta yang didalamnya terdapat titik *site* dan *route* dari lokasi pengguna menuju titik *site*. Tambahan menu yaitu menu

*help*, gunanya agar pengguna mengetahui cara kerja dari aplikasi ini.

### 3.3 Perancangan Aplikasi



Gambar 3.3 Flowchart Program

#### A. Menghubungkan hosting dengan android

Menghubungkan hosting dengan android bertujuan agar android dapat menampilkan data pada database dengan cara mengupload file PHP kedalam hosting. Perintah untuk menghubungkan sebagai berikut.

```
//Address of our scripts of the CRUD
public static final String
URL_ADD="http://thiara.esy.es/tiara/addData.php";
public static final String URL_GET_ALL =
"http://thiara.esy.es/tiara/getAllSite.php";
public static final String URL_GET_SITE =
"http://thiara.esy.es/tiara/getASite.php?speech=";
public static final String URL_UPDATE_EMP =
"http://thiara.esy.es/tiara/updateSite.php";
public static final String URL_DELETE_EMP =
"http://thiara.esy.es/tiara/deleteSite.php?id=";
```

#### B. Menghubungkan data baru ke dalam database

Untuk menampilkan data pada database kedalam android membutuhkan penghubung berupa bahasa pemrograman PHP. Bahasa PHP itu sendiri yaitu adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML atau web. Perintah untuk menghubungkan antara android dengan database yaitu melalui script sebagai berikut.

- Pada XML
 

```
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

```
android:text="Save Data"
android:onClick="saveBtn" />
```

- Pada Java**

```
public void saveBtn(View v)
{addData();}
//Adding an employee
private void addData()
{final String speech = speechET.getText().toString().trim();
final String siteid = siteidET.getText().toString().trim();
final String lon = lonET.getText().toString().trim();
final String lat = latET.getText().toString().trim();
class AddEmployee extends AsyncTask<Void,Void,String>
{ProgressDialog loading;
@Override
protected void onPreExecute()
{super.onPreExecute();
loading =
ProgressDialog.show(InsertDataActivity.this,"Adding...","Wait...",
false,false);}
@Override
protected void onPostExecute(String s)
{super.onPostExecute(s);
loading.dismiss();
Toast.makeText(InsertDataActivity.this, s,
Toast.LENGTH_LONG).show();}
@Override
protected String doInBackground(Void... v)
{HashMap<String,String> params = new HashMap<>();
params.put(Config.KEY_EMP_SPEECH, speech);
params.put(Config.KEY_EMP_SITEID, siteid);
params.put(Config.KEY_EMP_LONGITUDE, lon);
params.put(Config.KEY_EMP_LATITUDE, lat);
RequestHandler rh = new RequestHandler();
String res = rh.sendPostRequest(Config.URL_ADD, params);
return res;}
AddEmployee ae = new AddEmployee();
ae.execute();}
```

- Pada PHP**
  - Script PHP saat penambahan data**

```
<?php
if($_SERVER['REQUEST_METHOD']=='POST')
{
//Getting values
$speech = $_POST['speech'];
$siteid = $_POST['siteid'];
$longitude = $_POST['longitude'];
$latitude = $_POST['latitude'];
//Creating an sql query
$sql = "INSERT INTO db_site
(speech,siteid,longitude,latitude) VALUES
('$speech','$siteid','$longitude','$latitude)";
//Importing our db connection script
require_once('dbConnect.php');
//Executing query to database
if(mysqli_query($con,$sql))
{echo 'Data Berhasil Ditambahkan!;}
else
{echo 'Tidak Dapat Menambah Data!;}
//Closing the database
mysqli_close($con);}
?>
```
  - Script PHP untuk menuju database**

```
<?php
//Defining Constants
define('HOST','localhost');
define('USER','u344051002_bts');
define('PASS','tiara2828');
define('DB','u344051002_site');
//Connecting to Database
$con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
?>
```

**C. Mengeluarkan tombol speech to text**

Aplikasi ini menggunakan bahasa indonesia dalam mengenali suara. Menu tersebut berguna untuk memasukkan suara pengguna melalui tombol speech. Tombol tersebut akan muncul perintah “Ucapkan”, pengguna harus mengucapkan sesuai dengan speech yang sudah dimasukkan di dalam database, seperti mengucapkan “Malang 20(Dua puluh)” atau lain sebagainya. Perintah mengeluarkan tombol speech dengan script di bawah ini.

```
//method untuk speech recognition
protected static final int RESULT_SPEECH=1;
//tombol untuk menekan suara
public void voice(View v){
mendengarkan();}
private void mendengarkan(){
//panggil aplikasi bawaan android untuk merecognize speech
Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
//saya menggunakan parameter bahasa indonesia dalam mengenali suara
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "id-ID");
try {startActivityForResult(intent, RESULT_SPEECH);
//editTextSend.setText("");
} catch (ActivityNotFoundException a){
Toast t = Toast.makeText(getApplicationContext(), "Waduh devicemu
gak support", Toast.LENGTH_SHORT);t.show();}}
```

**D. Menghasilkan suara menjadi text**

Pengguna mengucapkan “Malang 20(Dua Puluh)” akan menghasilkan text berupa %22Malang%2020%22. Perintah untuk mengeluarkan hasil text, memerlukan script sebagai berikut.

```
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {resultCode, data);
switch(requestCode){
case RESULT_SPEECH:
{if(resultCode==RESULT_OK && null != data)
{ArrayList<String> text =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
speechDriHP = text.get(0);
speechDriHP = speechDriHP.replaceAll(" ", "%20");
speechDriHP = "%22" + speechDriHP + "%22";
textView.setText(speechDriHP);
getData(); } beak; } }
```

**E. Menghubungkan hasil text ke google map**



**Gambar 3.4 Tampilan Key**

Untuk menampilkan map pada android memerlukan penghubung berupa key atau kunci, yang bisa didapat melalui <https://console.developers.google.com> yang ditunjukkan pada gambar 4.1.

```
<string name="google_maps_key" translatable="false" templateMergeStrategy="preserve">
AIzaSyC1MPLepK9DxV0dAM6ImfUyF0gVQ1369Fk
</string>
```

**Gambar 3.5 Tampilan Key di Android Studio**

Pada Gambar 4.2 menunjukkan bahwa key yang telah di dapat, di tempatkan pada aplikasi Android Studio pada `google_maps_api.xml` yang bertujuan untuk menampilkan map pada android. Perintah untuk

menghubungkan hasil text ke map memerlukan script sebagai berikut.

```
private void tampilkanData(String json) {
try {JSONObject jsonObject = new JSONObject(json);
JSONArray result =
jsonObject.getJSONArray(Config.TAG_JSON_ARRAY);
JSONObject c = result.getJSONObject(0);
speechDriSever = c.getString(Config.TAG_speech);
siteidDariServer = c.getString(Config.TAG_siteid);
lonDariServer = c.getString(Config.TAG_longitude);
latDariServer = c.getString(Config.TAG_latitude);
//Toast.makeText(SpeechRecognitionKu.this, speechDriSever ,
Toast.LENGTH_LONG).show();
//textView.setText(speechDriSever);
if(!speechDriSever.equalsIgnoreCase("null"))
{Intent i = new Intent(SpeechRecognitionKu.this, MapsActivity.class);
i.putExtra("speech", speechDriSever);
i.putExtra("siteid", siteidDariServer);
i.putExtra("lon", lonDariServer);
i.putExtra("lat", latDariServer);
startActivity(i);}
else{Toast.makeText(SpeechRecognitionKu.this, "data tidak tersedia" ,
Toast.LENGTH_LONG).show();}
} catch (JSONException e)
{Toast.makeText(SpeechRecognitionKu.this, "Error atau Koneksi Tidak
Stabil" , Toast.LENGTH_LONG).show();
e.printStackTrace();}}
```

**F. Mengeluarkan lokasi awal dengan menggunakan GPS pada smartphone serta titik-titik site BTS**

Pengguna belum tentu berada disatu tempat tertentu, maka aplikasi ini mendukung keberadaan pengguna saat dimanapun dengan menggunakan GPS pada smartphone. Perintah untuk mengeluarkan GPS untuk aplikasi ini sebagai berikut.

```
private void addMarker() {
//googleMap.clear();
if(mapMarker != null) mapMarker.remove();
MarkerOptions options = new MarkerOptions();
// following four lines requires 'Google Maps Android API Utility
Library'
// https://developers.google.com/maps/documentation/android/utility/
// I have used this to display the time as title for location markers
// you can safely comment the following four lines but for this info
IconGenerator iconFactory = new IconGenerator(this);
iconFactory.setStyle(IconGenerator.STYLE_PURPLE);
options.icon(BitmapDescriptorFactory.fromBitmap(iconFactory.makeIcon("Lokasi saya")));
options.anchor(iconFactory.getAnchorU(), iconFactory.getAnchorV());
LatLng currentLatLng = new LatLng(mCurrentLocation.getLatitude(),
mCurrentLocation.getLongitude());
options.position(currentLatLng);
mapMarker = googleMap.addMarker(options);
long atTime = mCurrentLocation.getTime();
mLastUpdateTime = DateFormat.getTimeInstance().format(new
Date(atTime));
//mapMarker.setTitle(mLastUpdateTime);
//mapMarker.setTitle(mCurrentLocation.getLatitude() + "" +
mCurrentLocation.getLongitude());}
```

**G. Mengeluarkan rute**

Aplikasi ini menggunakan *rute* yang sudah disediakan oleh pihak google. Maka dalam hal ini penulis hanya memasukkan lokasi *site* BTS dan lokasi pengguna melalui GPS. Rute memerlukan parameter .Perintah untuk mengeluarkan *rute* untuk aplikasi ini sebagai berikut

```
@Override
public void onLocationChanged(Location location) {
Log.d(TAG, "Firing onLocationChanged.....");}
```

```
mCurrentLocation = location;
mLastUpdateTime = DateFormat.getTimeInstance().format(new
Date());
addMarker();
if(latSite != 0)
{//untuk rute
String url = getMapsApiDirectionsUrl();
ReadTask downloadTask = new ReadTask();
downloadTask.execute(url);}}
```

**H. Perhitungan jarak pada aplikasi**

Aplikasi ini memakai jarak yang mengambil dua titik secara garis lurus, disebabkan untuk pencarian jarak terpendek dari lokasi pengguna menuju *site* BTS. Perintah untuk mengeluarkan *rute* untuk aplikasi ini sebagai berikut.

```
//membuat array jarak
float[] jarak = new float[1];
//lat gps, lon gps, lat site, lon site, variabel penyimpan
Location.distanceBetween(mCurrentLocation.getLatitude(),
mCurrentLocation.getLongitude(), latSite, lonSite, jarak);
if(jarak[0]/1000 < 1) //kalau hasilnya kurang dari 1 km
{mapMarker.setTitle("ke tujuan " + (jarak[0]/1000) + " km");}
else
{mapMarker.setTitle("ke tujuan " + (int)(jarak[0]/1000) + " km");}
```

```
////////// untuk jarak ////////////
Log.d(TAG, "Marker added.....");
googleMap.moveCamera(CameraUpdateFactory.newLatLng(currentLat
Lng));
Log.d(TAG, "Zoom done.....");
siteMarker = googleMap.addMarker(new MarkerOptions().position(new
LatLng(latSite,lonSite)).title(siteidDariSebelumnya));}
```

**IV.HASIL PENGUJIAN DAN PEMBAHASAN**

**4.1 Pengujian**

**4.1.1 Pengujian Speech To Text Terhadap Kebisingan**

➤ **Langkah Pengujian**

Langkah-langkah pengujian *speech to text* yaitu :

1. Posisikan perangkat *smartphone* di depan pengguna.
2. Perangkat dan pengguna dalam posisi tegak lurus dengan jarak 10 cm.
3. Pengguna mengucapkan “Malang 20 (dua puluh)” sebanyak 10 kali pengucapan.
4. Setiap respon aplikasi antara berhasil atau tidak di catat.
5. Pengujian untuk jarak 35 cm ikuti kembali langkah 1 sampai 4.

**A. Pengujian di Dalam Ruangan dengan Tingkat Kebisingan 63 dB**

**Tabel 4.1 Hasil Pengujian Speech To Text di Dalam Ruangan**

Kebisingan yang timbul	Nama Pengguna	Jarak Perangkat dan Pengguna			
		10 cm		35 cm	
		Berhasil	Gagal	Berhasil	Gagal
63 dB (Dalam ruangan)	Aisy	9	1	4	6
	Linda	9	1	5	5
	Agnes	8	2	8	2
	Rizal	10	0	9	1
	Rezsa	8	2	9	1
Jumlah		44	6	35	15
Persentase (%)		88%	12%	70%	30%

**B. Pengujian di Luar Ruangan dengan Tingkat Kebisingan 70 dB**

**Tabel 4.2 Hasil Pengujian Speech To Text di Luar Ruangan**

Kebisingan yang timbul	Nama Pengguna	Jarak Perangkat dan Pengguna			
		10 cm		35 cm	
		Berhasil	Gagal	Berhasil	Gagal
70 dB (Luar ruangan)	Aisy	9	1	5	5
	Linda	8	2	4	6
	Agnes	8	2	6	4
	Rizal	9	1	8	2
	Rezza	9	1	9	1
Jumlah		43	7	32	18
Presentase (%)		86%	14%	64%	36%

**C. Pengujian di Sekitar Jalan Raya dengan Tingkat Kebisingan 75 dB**

**Tabel 4.3 Hasil Pengujian Speech To Text di Sekitar Jalan Raya**

Kebisingan yang timbul	Nama Pengguna	Jarak Perangkat dan Pengguna			
		10 cm		35 cm	
		Berhasil	Gagal	Berhasil	Gagal
75 dB (Sekitar jln raya)	Aisy	8	2	4	6
	Linda	8	2	4	6
	Agnes	8	2	5	5
	Rizal	9	1	9	1
	Rezza	9	1	8	2
Jumlah		42	8	30	20
Presentase (%)		84%	16%	60%	40%

**4.1.2 Pengujian Speech To Text Menuju Maps**

**Tabel 4.4 Hasil Pengujian Speech To Text Menuju Maps**

No	Nama	Intensitas Suara (Db)	Commen voice	Waktu hasil speech to text ke maps	Rata-rata Waktu
1	Aisy	65 dB	Malang 20	1:17	1:75
			Malang 58	1:48	
			Malang 60	94 detik	
			Batu 77	1:89	
			Batu 90	1:39	
2	Linda	66 dB	Malang 20	1:18	1:70
			Malang 58	1:32	
			Malang 60	1:60	
			Batu 77	1:72	
			Batu 90	1:25	
3	Agnes	66 dB	Malang 20	1:78	1:71
			Malang 58	1:98	
			Malang 60	1:31	
			Batu 77	58 detik	
			Batu 90	1:10	
4	Rizal	70 dB	Malang 20	1:56	1:51
			Malang 58	1:14	
			Malang 60	59 detik	
			Batu 77	68 detik	
			Batu 90	1:78	
5	Rezza	69 db	Malang 20	1:17	1:68
			Malang 58	85 detik	
			Malang 60	1:71	
			Batu 77	1:35	
			Batu 90	1:57	

**4.1.3 Pengujian dan Perhitungan Jarak**

**Tabel 4.5 Hasil Pengujian Jarak pada Aplikasi dan Perhitungan Rumus Euclidean Distane**

No	Titik Awal dan Akhir	Titik Awal (x)		Titik Akhir (y)		Jarak pada aplikasi [Km]	Hasil Perhitungan [Km]
		Latitudel	Longitudel	Latitude2	Longitude2		
1	POLINEMA-20MLG020	-7,947481	112,616928	-7,95233095	112,6170033	0,54418606	0,539956652
2	POLINEMA-20MLG038	-7,947481	112,616928	-7,947153	112,619594	0,27375013	0,299014107
3	POLINEMA-20MLG060	-7,947481	112,616928	-7,94052778	112,6103333	1	1,06679
4	POLINEMA-20MLG039	-7,947481	112,616928	-7,95703056	112,6185889	1	1,079006086
5	POLINEMA-20MLG055	-7,947481	112,616928	-7,97154	112,62067	2	2,710424595
6	Balai Kota-20MLG003	-7,977766	112,633895	-7,94645	112,604	4	4,819486498
7	Balai Kota-20MLG021	-7,977766	112,633895	-7,9491	112,6455	3	3,442647983
8	Balai Kota-20MLG057	-7,977766	112,633895	-7,94342	112,62411	3	3,975497375
9	Balai Kota-20MLG033	-7,977766	112,633895	-7,99194285	112,6334682	1	1,578867772
10	Balai Kota-20MLG026	-7,977766	112,633895	-7,9693894	112,6318548	0,9528397	0,959734056
11	Rumah Saya-20BTU077	-7,899299	112,691047	-7,91567973	112,6249135	7	7,584385057
12	Rumah Saya-20BTU090	-7,899299	112,691047	-7,89761998	112,6737406	1	1,935576504
13	Rumah Saya-20BTU003	-7,899299	112,691047	-7,91531	112,656	4	4,289241589
14	Rumah Saya-20BTU132	-7,899299	112,691047	-7,8815582	112,6768589	2	2,528775109
15	Rumah Saya-20BTU134	-7,899299	112,691047	-7,86033923	112,6830756	4	4,426812326
16	Giant Singosari-20BTU105	-7,915461	112,655265	-7,93022989	112,6429732	2	2,13897209
17	Giant Singosari-20BTU006	-7,915461	112,655265	-7,87505	112,666	4	4,654530999
18	Giant Singosari-20BTU078	-7,915461	112,655265	-7,87702778	112,5784167	9	9,564870105
19	Giant Singosari-20BTU007	-7,915461	112,655265	-7,8981	112,601	6	6,342345241
20	Giant Singosari-20BTU012	-7,915461	112,655265	-7,84354	112,699	9	9,370243658

**V. PENUTUP**

**5.1 Kesimpulan**

- Hasil pengujian tampilan menu menyatakan bahwa aplikasi telah sesuai dengan yang diharapkan, serta hasil pengujian terhadap perangkat yang berbeda telah sesuai yang dengan diharapkan dan tidak ada error saat menjalankan *insert menu*, *speech*, *maps* maupun jarak pada perangkat lain. Hasil pengujian *speech to text* antara laki-laki dan perempuan diperoleh suara laki-laki lebih banyak keberhasilannya dari pada perempuan, diakibatkan besar suara laki-laki lebih besar dari pada perempuan. Presentase dari *speech to text* diperoleh bahwa aplikasi ini dapat bekerja dengan baik, dapat dilihat dari tingkat keberhasilannya yaitu rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 63dB sebanyak 79% keberhasilannya, rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 70dB sebanyak 75% keberhasilannya, rata-rata untuk jarak 10cm dan 35cm dengan tingkat kebisingan 75dB sebanyak 72% keberhasilannya, maka keberhasilan lebih akurat pada tingkat kebisingan 63dB dengan tingkat keberhasilannya 79%. Hasil pengujian waktu saat membuka *maps* dari hasil *text* ke *maps*, lebih cepat laki-laki yaitu 51 detik dan 1 menit 68 detik daripada perempuan disebabkan intensitas suara laki-laki lebih besar.
- Perbandingan jarak pada aplikasi yang dibuat dengan perhitungan melalui rumus Euclidean Distance diperoleh rata-rata untuk jarak pada aplikasi yang dibuat yaitu 3,65345287 kmdan hasil perhitungan dengan rumus Euclidean Distance yaitu 3,66535889 km.

#### DAFTAR PUSTAKA

- [1] Permatasari Riski. 2014. *Aplikasi Informasi Rute Angkutan Kota (Angkot) Di Kota Malang Pada Perangkat Android*. Malang: POLINEMA.
- [2] Hayatun, Noviatus. 2015. *Rancang Bangun Aplikasi Al Quran Digital Untuk Penyandang Disabilitas Tangan Berbasis Command Voice Pada Perangkat Android*. Malang: POLINEMA.
- [3] Setiawan. 2013. *Application-in-world*. Surabaya: STIKOM Surabaya.
- [4] Ichwan, Fifin. 2011. *Pengukuran Kinerja Goodreads Application Programming Interface (API) Pada Aplikasi Mobile Android*. Bandung: ITB-Bandung.
- [5] Budijono. 2013. *Simulasi Keandalan Base Transceiver Station (BTS) Dengan Menggunakan Program Visual Basic*. Diponegoro: Universitas Diponegoro.
- [6] Taylor, Paul. 2007. *Speech-to-text Synthesis*. United Kingdom: University Pree.
- [7] Rian, Emha. 2015. *Rancang Bangun Pembuatan Aplikasi "Voice Recognition Secure" Sebagai Media Keamanan Data Berbasis Android*. Yogyakarta: STMIK-AMIKOM.
- [8] Artikel "Global Positioning System (Gps) Overview". Diakses melalui [https://www.academia.edu/4915337/GLOBAL\\_POSITIONING\\_SYSTEM\\_GPS\\_OVERVIEW](https://www.academia.edu/4915337/GLOBAL_POSITIONING_SYSTEM_GPS_OVERVIEW) pada tanggal 20 Mei 2016 pukul 19.00.
- [9] Weinberg, S. *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*. New York: Wiley, p. 7, 1972.