

**NASKAH PUBLIKASI**

**PROYEK TUGAS AKHIR**

**IMPLEMENTASI METODE HAVERSINE FORMULA DALAM APLIKASI  
UNTUK MENENTUKAN LOKASI EMERGENCY SERVICE TERDEKAT DI  
DAERAH ISTIMEWA YOGYAKARTA**

Program Studi Teknik Informatika

Disusun oleh

**JATRA SULISTIO**

**3125111202**

Kepada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2019**

**NASKAH PUBLIKASI**

**IMPLEMENTASI METODE HAVERSINE FORMULA DALAM APLIKASI  
UNTUK MENENTUKAN LOKASI EMERGENCY SERVICE TERDEKAT DI  
DAERAH ISTIMEWA YOGYAKARTA**

Disusun oleh:

**Jatra Sulistio**

**3125111202**

Telah disetujui oleh pembimbing

Pembimbing

Yuli Asriningtias, S.Kom., M.Kom

Tanggal:

# IMPLEMENTASI METODE HAVERSINE FORMULA DALAM APLIKASI UNTUK MENENTUKAN LOKASI EMERGENCY SERVICE TERDEKAT DI DAERAH ISTIMEWA YOGYAKARTA

**Jatra Sulistio**

*Information Systems Studies Program, Faculty of Information & Electrical Technology  
University of Tecnology Yogyakarta  
Jln. Ring Road Utara No.81 Sleman DIY  
Email: [jatra691@gmail.com](mailto:jatra691@gmail.com)*

## ABSTRAK

Yogyakarta memiliki banyak tempat wisata yang mengundang banyak wisatawan domestic maupun mancanegara, yang juga memicu pertumbuhan Rumah Sakit, puskesmas, klinik, dan kantor polisi. Dengan semakin banyaknya Klinik, puskesmas, rumahsakit maupun kantor polisi yang ada di Yogyakarta, wisatawan sering kebingungan saat mencari lokasi Emergency yang terdekat, apalagi di daerah yang belum di kenal. Informasi Puskesmas, Klinik, Rumah Sakit dan Kantor Polisi di internet sebenarnya sudah banyak, tapi informasi tersebut berupa review dan alamat yang tidak banyak membantu saat berada di lokasi yang tidak dikenal. Saat ini handphone sudah populer digunakan, memiliki fitur-fitur yang dapat dimanfaatkan untuk mencari lokasi saat terjadi emergency, seperti GPS dan mobile browser yang telah mendukung geolocation. Dua fitur tersebut memungkinkan dapat mengetahui koordinat pengguna handphone secara otomatis melalui aplikasi android. Aplikasi Mobile Emergency Service adalah suatu aplikasi yang mempunyai fungsi untuk menampilkan keberadaan lokasi Emergency Service terdekat yang terdapat di Daerah Istimewa Yogyakarta, dengan metode haversine formula dan koneksi internet. Aplikasi Emergency Service ini dapat disimpulkan sebagai sarana untuk menjawab atau mengatasi beberapa masalah bagi parawisatawan dalam mencari lokasi Emergency diantaranya memudahkan pengguna/masyarakat saat keadaan darurat dalam mencari lokasi Emergency Service aplikasi ini mencakup emergency Daerah Istimewa Yogyakarta.

**Kata Kunci:** Emergency Service, Haversine Formula, Android, Yogyakarta.

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi sangatlah cepat berkembang, secara tidak di sengaja telah meningkatkan kebutuhan masyarakat dan sekaligus telah mengubah gaya hidup masyarakat. Salah satu teknologi yang sedang berkembang adalah teknologi *handphone*, kini masyarakat tidak hanya butuh satu *handphone* yang hanya bisa digunakan untuk berkomunikasi seperti menerima telpon ataupun mengirim pesan singkat, namun *handphone* diharapkan juga bisa dimanfaatkan untuk hal-hal penting lainnya, seperti mencari informasi lokasi.

Kota Yogyakarta adalah salah satu kota tujuan wisata yang ada di Indonesia, selain menawarkan wisata budaya, Yogyakarta juga banyak menawarkan wisata-wisata lainnya, seperti wisata kuliner, pantai, candi, bahkan bahkan wisata modern seperti *water boom*, taman lampion dll. Saat berwisata ke Yogyakarta, tidak

sedikit wisatawan yang mengalami hal yang tidak diinginkan, seperti kecelakaan, kehilangan, dll. Dari kejadian ini, menimbulkan masalah bagi para wisatawan dalam mencari atau menentukan *emergency service* terdekat dengan posisinya saat itu sehingga dapat menghemat waktu dan mendapatkan penanganan secepatnya.

Hal mudah untuk menyelesaikan permasalahan tersebut yakni dengan membangun sebuah aplikasi yang dapat melakukan pencarian *emergency service* terdekat dengan lokasi pengguna (*current location*). Solusi yang di tawarkan akan di wujudkan dengan menggunakan *Haversine Formula*, *Haversine Formula* merupakan sebuah persamaan yang memberikan jarak lingkaran besar (radius) antara dua titik pada permukaan bola (bumi) berdasarkan garis bujur dan lintang. *Haversine Formula* adalah rumus yang tepat untuk menghitung jarak antara dua titik yakni dengan inputan *latitude* dan *longitude*,

sebagai titik awal dan akhir, maka akan di hitung jarak antara titik-titik yang berada di dekatnya, output yang di tampilkan adalah *emergency service* terdekatnya.

Dengan demikian, maka penulis bermaksud untuk merancang dan membuat aplikasi *emergency service* berbasis android dengan menggunakan android studio dan MySQL sehingga penulis mengangkat judul tugas akhir “Menentukan Rute Terdekat Emergency Service Daerah Istimewa Yogyakarta Berbasis Android Dengan Metode Haversine Formula”. Aplikasi ini di harapkan akan membantu para wisatawan maupun masyarakat yogyakarta untuk mencari lokasi *emergency service* terdekat.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka penulis membuat rumusan masalah bagaimana membuat aplikasi *Emergency Service* di Daerah Yogyakarta yang efektif dan cepat serta mudah dibawa ketika dibutuhkan?

## 1.3 Batasan Masalah

Dalam pengajuan proposal tugas akhir penulis membahas beberapa masalah antara lain :

- Emergency service* ini meliputi Puskemas, Klinik, Rumah sakit dan Kantor Polisi yang ada di Daerah Istimewa Yogyakarta.
- Sistem menggunakan peta dan rute dari Google Maps.
- Sistem hanya mencakup daerah kota Yogyakarta.
- Haversine formula hanya menarik garis lurus untuk melihat jarak *emergency service* terdekat dengan user.

## 1.4 Tujuan penelitian

Tujuan dari tugas akhir ini antara lain :

- Memudahkan wisatawan maupun masyarakat mendapat informasi lokasi terdekat *Emergency Service* di daerah kota Yogyakarta.
- Memudahkan wisatawan maupun masyarakat dalam menentukan *Emergency Service* yang tepat.

## 2. KAJIAN PUSTAKA DAN TEORI

### 2.1 Kajian Pustaka

penelitian tentang “Aplikasi Pencarian Tempat Wisata Berbasis GPS dengan Metode Radius dan Rating” Dari hasil penelitian tugas akhir tersebut dapat ditarik kesimpulan bahwa aplikasi memberi kemudahan dalam mencari lokasi wisata berdasarkan radius wisatawan dan rating, memberikan rute, dan deskripsi tempat wisata. [1]

penelitian tentang “Pemanfaatan Location Based Service Berbasis Android untuk Pemetaan Kantor Polisi Wilayah Kota Pekanbaru Riau” Dengan menggunakan Sistem berbasis LBS ini masyarakat dapat mencari dan mengakses informasi kantor polisi yang ada di wilayah kota Pekanbaru dengan memanfaatkan layanan Google Map API yang telah disediakan untuk mengembangkan aplikasi. [2]

melakukan penelitian tentang “Menentukan jarak terdekat hotel dengan metode haversine formula.” dimana dalam penelitian tersebut dapat membantu dan memudahkan wisatawan untuk mencari hotel terdekat yang ada di kota malang. [3]

## 2.2 Landasan Teori

### 2.2.1 Implementasi

Implementasi adalah bermuara pada aktivitas, aksi, tindakan, atau adanya mekanisme suatu sistem. Implementasi bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan [4]

### 2.2.2 Metode

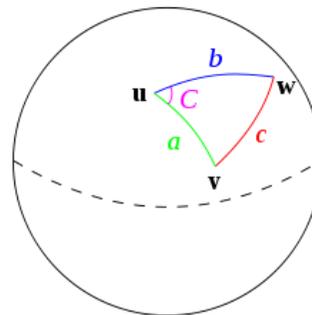
Metode berasal dari kata Yunani *meta* + *hodos* = jalan, *methodos* = jalan sampai. Metode adalah cara teratur yang digunakan untuk melaksanakan suatu pekerjaan agar tercapai sesuai dengan yang dikehendaki; cara kerja yang sistematis untuk memudahkan pelaksanaan suatu kegiatan guna mencapai tujuan yang ditentukan.

### 2.2.3 Haversine Formula

Teorema Haversine Formula adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari *longitude* dan *latitude*. Ini merupakan bentuk persamaan khusus dari trigonometri bola, *law of haversines*, mencari hubungan sisi dan sudut pada garis segitiga dalam bidang bola.

Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Rios di tahun 1801. Istilah *haversine* ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Rios menggunakan *haversine* pertama kali dalam penelitiannya tentang “Masalah Utama Astronom Nautical”, Proc.Royal Soc, Dec 22. 1796. *Haversine* digunakan untuk menemukan jarak antar bintang.

#### 2.2.3.1 Hukum Haversine



**Gambar 2.1** Segitiga bola diselesaikan dengan hukum haversine formula

Hukum *Haversine* adalah sebuah persamaan yang digunakan berdasarkan bentuk bumi yang bulat (*spherical earth*) dengan menghilangkan faktor bahwa bumi itu sedikit elips (*elipsoidal factor*). Ini merupakan kasus khusus dari formula umum dalam trigonometri bola, hukum *haversine*, yang berkaitan dengan sisi dan sudut segitiga bola. Dalam unit bola,

sebuah “segitiga” pada permukaan bola didefinisikan sebagai lingkaran-lingkaran besar yang menghubungkan tiga poin **u**, **v**, dan **w** pada bola. Jika panjang dari ketiga sisi, **a** adalah (dari **u** ke **v**), **b** (dari **u** ke **w**), dan **c** (dari **v** ke **w**), dan sudut sudut yang berlawanan **c** adalah **C**. maka hukum *haversine* menjadi:

Haversine (c) = haversine(a-b) + cos(a) cos(b) haversine (C).

Rumus haversin sendiri adalah

$$\text{haversine}(\emptyset) = \sin^2\left(\frac{\emptyset}{2}\right) = \frac{1}{2} (1 - \cos(\emptyset))$$

(3)

Dimana sudut ( $\emptyset$ ) = sudut (C) pada gambar 2.1 maka :

$$\text{haversine}(C) = \sin^2\left(\frac{C}{2}\right) = \frac{1}{2} (1 - \cos c)$$

Perhatikan gambar 2.1 C adalah sudut yang terbentuk dari sisi a dan b, sedangkan sisi yang terletak di depan titik u memiliki panjang c.

Berdasarkan definisi tersebut, di peroleh :

$$\cos c = 1 - 2 \text{ haversine } C$$

(4)

Dengan memperhatikan gambar 2.1 bahwa besar C=c dan u adalah titik dengan besar sudut C, maka dapat di tulis :

$$\cos u = 1 - 2 \text{ haversine } u$$

(5)

$$\cos C = 1 - 2 \text{ haversine } C$$

(6)

Selanjutnya berdasarkan aturan cosinus pada segitiga bola yaitu :

$$\cos C = \cos a \cos b + \sin a \sin b \cos u$$

(7)

Maka dari persamaan (7) dengan persamaan (6) dan persamaan (5) di dapatkan.

$$\begin{aligned} \cos C &= \cos a \cos b + \sin a \sin b \cos u \\ 1 - 2 \text{ haversine } C &= \cos a \cos b + \sin a \sin b (1 - 2 \text{ haversine } u) \end{aligned}$$

$$1 - 2 \text{ haversine } C = \cos a \cos b + \sin a \sin b - 2 \sin a \sin b \text{ haversine } u$$

(8)

Karena nilai  $\cos a \cos b + \sin a \sin b = \cos(a - b)$  maka persamaan 8 menjadi

$$1 - 2 \text{ hav } C = \cos(a - b) - 2 \sin a \sin b \text{ hav } u$$

(9)

Berdasarkan persamaan (4) maka persamaan (9) menjadi

$$1 - 2 \text{ haversine } C = 1 - 2 \text{ haversine}(a - b) - 2 \sin a \sin b \text{ haversine } u$$

$$-2 \text{ haversine } C = \cos(a - b) - 2 \sin a \sin b \text{ haversine } u - 1$$

$$-2 \text{ haversine } C = 1 - 2 \text{ haversine}(a - b) - 2 \sin a \sin b \text{ haversine } u - 1$$

$$-2 \text{ haversine } C = -2(a - b) - 2 \sin a \sin b \text{ haversine } u$$

Kalikan kedua ruas dengan  $(-\frac{1}{2})$  sehingga di peroleh

$$\text{haversine } C = \cos(a - b) + \sin a \sin b \text{ haversine } u$$

(10)

berdasarkan rumus busur, untuk menghitung sudut C pada gambar 2.1 adalah :

$$C = \frac{c}{R}$$

(11)

Berdasarkan persamaan (11) maka persamaan (9) menjadi :

$$\text{Haversine } \frac{c}{R} = \cos(a - b) + \sin a \sin b \text{ haversine } u$$

$$\text{Haversine } c = R * \cos(a - b) + \sin a \sin b \text{ haversine } u$$

$$\text{Haversine } c = R * \text{ haversine } C$$

(12)

Untuk menyelesaikan jarak ke c, terapkan *archaversine* (invers *haversine*) ke  $C = \text{haversine}(C)$  atau gunakan fungsi *arcsin* (sinus terbalik).

$$\begin{aligned} c &= R \text{ archav}(C) = 2R \arcsin(\sqrt{C}) \\ \frac{c}{R} &= \frac{2R \arcsin(\sqrt{C})}{2R} \\ \sqrt{\text{hav}(a_2 - a_1) + \cos a_1 \cos a_2 \text{ hav}(b_2 - b_1)} &= \arcsin \end{aligned}$$

(13)

Dimana :

$$\cos c = \cos(a) \cos(b) + \sin(a) \sin(b) \cos(C)$$

$$\text{haversine}(C) = \text{haversine}(a_2 - a_1) + \cos a_1 \cos a_2 \text{ haversine}(b_2 - b_1)$$

Panjang jari-jari lingkaran  $R = uv = uw$  atau radius bumi =

$$a = \text{jarak } u \text{ ke } v \text{ atau jari-jari} = \text{latitude } 1 = a_1$$

$$b = \text{jarak } u \text{ ke } w \text{ atau jari-jari} = \text{latitude } 2 = a_2$$

$u =$  titik yang berlawanan dengan sisi  $c =$  longitude =  $b_1$  adalah longitude 1 dan  $b_2$  longitude 2

$c =$  jarak titik  $v$  ke  $w$  atau jarak

$$C = \frac{c}{r} = \text{sudut antara sisi } u \text{ ke } v \text{ dan sisi } u \text{ ke } w$$

Haversine formula nantinya akan di gunakan dalam perhitungan jarak antara dua titik GPS. Dalam hal ini adalah titik GPS *user* dan titik GPS tujuan, titik GPS ini berisikan *latitude* dan *longitude*, sehingga dapat menjadi kunci utama dalam perbandingan jarak pada penentuan lokasi *emergency service* terdekat. Berikut rumus Haversine yang akan di implementasikan kedalam tugas akhir ini.

$$\frac{c}{R} = \frac{2R * \arcsin(\sqrt{\sin^2(\frac{a_2 - a_1}{2}) + \cos a_1 \cos a_2 \sin^2(\frac{b_2 - b_1}{2})}}{2R}$$

(14)

Dimana :

$$\text{Haversine } (\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos(\theta)}{2}$$

$a_1 = \theta_1$  = latitude dari titik 1 atau posisi user

$a_2 = \theta_2$  = latitude dari titik 2 atau lokasi tujuan

$b_1 = \lambda_1$  = longitude dari titik 1 atau posisi user

$b_2 = \lambda_2$  = longitude dari titik 2 atau posisi tujuan

$c = d$  = jarak antara dua titik

$R$  = radius bumi = 6371km

Maka rumus (14) dapat kita tulis dengan :

$$d = \frac{2R * \arcsin\left(\sqrt{\sin^2\left(\frac{\theta_2 - \theta_1}{2}\right) + \cos \theta_1 \cos \theta_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)}{1} \quad (15)$$

untuk mempermudah penulisan rumus, rumus (15) dapat kita sederhanakan lagi seperti pada rumus (16).

$$d = R * 2 * \arcsin(\sqrt{a + c}) \quad (16)$$

Dimana :

$d$  = jarak antara dua titik

$R$  = radius bumi = 6371km

$\arcsin$  = arka sinus

$$a = \sin^2\left(\frac{\Delta lat}{2}\right)$$

$$c = \cos \theta_1 \cos \theta_2 \sin^2\left(\frac{\Delta long}{2}\right)$$

$$\Delta lat = \theta_2 - \theta_1$$

$$\Delta long = \lambda_2 - \lambda_1$$

$$1 \text{ derajat} = 1 \frac{\pi}{180} \text{ radian} = 0,0174532925 \text{ radian} \quad (17)$$

Jadi, sebelum kita masukan latitude dan longitude ke dalam rumus, kita harus mengubahnya dulu kedalam bentuk radian, seperti pada rumus (17).

### 2.2.4 Aplikasi

aplikasi adalah sistem yang dirancang dan disusun sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan sarana komputer sebagai sarana penunjangnya.[5]

### 2.2.5 Lokasi

definisi Lokasi (*Location*) adalah posisi pasti dalam ruang. Dalam Geografi lokasi mempunyai dua makna yaitu lokasi *absolut* dan lokasi relatif. [6]

- Lokasi *absolut* adalah lokasi di permukaan yang ditentukan oleh sistem koordinat garis lintang dan garis bujur, disebut juga lokasi mutlak.
- Lokasi relatif adalah lokasi sesuatu objek yang nilainya ditentukan oleh objek-objek lain di luarnya

### 2.2.6 Emergency Service

Layanan darurat atau *emergency services* menurut FEMA (Federal Emergency Management Agency) adalah kejadian yang tidak direncanakan dan tidak diinginkan bias mengakibatkan kematian atau luka serius pada pegawai, pelanggan, atau bahkan masyarakat, mematikan/mengganggu pekerjaan,

menyebabkan kerukana fisik atau mengancam kerusakan fasilitas bangunan, atau merusak citra public.

### 2.2.7 Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. [7] Beberapa pengertian lain dari Android yaitu :

- Merupakan *platform* terbuka (*open source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
- Merupakan sistem operasi yang dibeli **Google Inc** dari **Android Inc**.
- Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time environment* yang disebut **DVM** (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil.

### 2.2.8 Google Maps

*Google Maps* adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. [8]

### 2.2.9 Global Positioning System (GPS)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. [9]

### 2.2.10 Web Service

*Web Service* adalah sekumpulan *application logic* beserta objek-objek dan metode-metode yang dimilikinya yang terletak di suatu server yang terhubung ke internet sehingga dapat diakses menggunakan *protocol* HTTP dan SOAP (*Simple Object Access Protocol*). [10]

### 2.2.11 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 – Desember 1999. [10]

### 2.2.12 PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skripsi HTML. [11]

### 2.2.13 Android Studio

Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip

dengan Eclipse disertai dengan ADT plugin (Android Development Tools). [10]

### 2.2.14 Java Development Kit (JDK)

*Java Development Kit (JDK)* adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java. [10]

### 2.2.15 Unified Modeling Language (UML)

*Unified Modeling Language (UML)* merupakan perangkat atau alat dan proses yang digunakan dalam pemodelan spesifikasi kebutuhan pengguna yang bernuansa *object oriented*. [11]

### 2.2.16 Use Case

*Use case* merupakan sejumlah skenario yang dapat digunakan untuk mengidentifikasi sejumlah penggunaan sistem yang mungkin ada atau terjadi, dan menyediakan deskripsi yang rinci tentang bagaimana sistem akan digunakan. Pada intinya *use case* sebenarnya mengatakan cerita tentang bagaimana seorang pengguna akhir (yang memainkan satu dari sejumlah peran yang mungkin) yang berinteraksi dengan sistem yang berada dibawah sejumlah situasi dan kondisi yang sifatnya spesifik. [11]

### 2.2.17 Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, activity diagram memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah activity diagram mendukung behavior paralel. Node pada sebuah activity diagram disebut sebagai action, sehingga diagram tersebut menampilkan sebuah activity yang tersusun dari action.

### 2.2.18 State Diagram

Diagram *state* memperlihatkan bagaimana sistem/perangkat lunak akan menanggapi *event-event* saat ia bergerak disepanjang *state* (keadaan) yang didefinisikan pada peringkat ini.

### 2.2.19 Sequence Diagram

Sequence diagram adalah grafik duadimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan lifeline ditunjukkan dalam dimensi vertikal.

### 2.2.20 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram (ERD)* dapat mengekspresikan struktur logis dari *database* grafis secara keseluruhan. ERD yang sederhana dan jelas kualitasnya sebagian besar digunakan untuk menjelaskan penggunaan *Entity Relationship Model (ERM)*.

## 3. METODE PENELITIAN

### 3.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh hasil data yang akurat dan valid. Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

- a. Observasi
- b. Studi Pustaka
- c. Analisis Data

## 3.2 Pengembangan Sistem

### 3.2.1 Desain Perancangan

Dalam perancangan sistem pemetaan lokasi *Emergency Service* ini terbagi menjadi 3 bagian yaitu :

1. Perancangan sistem
2. Perancangan basis data
3. Perancangan *interface*

### 3.3 Implementasi

Sistem ini akan diimplementasikan pada user pengguna android yang akan melakukan pencarian *emergency service*.

### 3.3.1 Kebutuhan Nonfungsional Sistem

Merupakan kebutuhan nonfungsional mengenai kebutuhan pendukung sistem yang akan dibuat untuk memenuhi kebutuhan sistem yang meliputi kebutuhan *hardware* dan *software*. Kebutuhan nonfungsional tersebut adalah :

- a. Hardware
- b. Software

### 3.4 Pengujian

Metode ini merupakan tahap pengujian sistem yang dilakukan sistem diimplementasikan. Pengujian digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.

## 4. ANALIS DAN PERANCANGAN SISTEM

### 4.1 Analisis Sistem

Berdasarkan dari analisa terhadap aplikasi sejenis yang sudah ada, aplikasi yang dibangun ini dapat menunjukkan letak lokasi *emergency service* yang ada di Yogyakarta berdasarkan posisi letak *user* yang sudah terintegrasi dengan GPS serta koneksi internet dan mempunyai tampilan berupa *map* yang terintegrasi dengan *google maps*.

### 4.2 Perancangan Sistem

Rancangan sistem secara umum dilakukan dengan maksud untuk memberikan gambaran umum tentang sistem yang baru atau sistem yang akan diusulkan. Rancangan ini mengidentifikasi komponen-komponen sistem informasi yang akan dibangun secara rinci.

### 4.3 Rencana Pengembangan Sistem

Tahap desain sistem dilakukan setelah tahap analisis sistem. Desain sistem menggunakan notasi untuk membantu dalam memberikan gambaran atau penjelasan pada pemakai tentang sistem yang akan di buat, yaitu mengenai aliran dan informasi data yang telah di inputkan sampai dengan data yang akan ditampilkan nantinya.

### 4.4 Struktur Basisdata

Dalam perancangan basisdata digunakan tabel-tabel untuk penyimpanan data-data yang diperlukan, tabel-tabel tersebut disimpan dalam bentuk file. Rincian file tersebut adalah sebagai berikut :

**Tabel 4.1** Struktur tabel Bpbd

Field	Type	Extra
Nama Tabel	Bpbd	
Primary Key	Id_bpbd	

Id_bpbd	varchar(10)	auto_increment
Nama_bpbd	varchar(100)	
Alamat	varchar(100)	
Tlpn	Int	
Latitude	Double	
Longitude	Double	
Profil	Text	
Galeri	varchar(10000)	

**Tabel 4.2** Struktur tabel Polisi

Nama Tabel	Pilisi	
Primary Key	Id_polisi	
<b>Field</b>	<b>Type</b>	<b>Extra</b>
Id_polisi	varchar(10)	auto_increment
Nama_polisi	varchar(100)	
Alamat	varchar(100)	
Tlpn	Int	
Latitude	Double	
Longitude	Double	
Profil	Text	
Galeri	varchar(10000)	

**Tabel 4.3** Struktur tabel Kesehatan

Nama Tabel	kesehatan	
Primary Key	Id_kesehatan	
<b>Field</b>	<b>Type</b>	<b>Extra</b>
Id_kesehatan	varchar(10)	auto_increment
Nama_kesehatan	varchar(100)	
Alamat	varchar(100)	
Latitude	Double	
Longitude	Double	
Profil	Text	
Galeri	varchar(10000)	

## 5. IMPLEMENTASI SISTEM

Implementasi dari analisis dan perancangan pada penelitian ini merupakan sebuah pembuatan aplikasi Android *emergency service* yang ada di Daerah Istimewa Yogyakarta. Aplikasi Android ini bermanfaat untuk memberikan kemudahan dalam mendapatkan informasi *emergency service* seputar Daerah Istimewa Yogyakarta yang dilengkapi dengan tampilan *Maps* dan *Haversine Formula* yang menunjukkan rute menuju lokasi yang ada di Daerah Istimewa Yogyakarta.

### 5.1 Implementasi

Tujuan implementasi sistem adalah untuk menjelaskan manual modul kepada semua *user* yang akan menggunakan sistem ini. Sehingga pengguna dapat merespon apa yang ditampilkan di sistem dan memberikan masukan kepada pembuat sistem untuk dilakukan perbaikan agar sistem menjadi lebih baik.

Dalam menerapkan racangan yang telah dibuat, ada beberapa hal yang harus dibutuhkan. Perangkat keras dan perangkat lunak merupakan dua hal yang selalu dibutuhkan dalam mengimplementasi racangan yang telah ada.

Sebelum di lakukan implementasi pada program, akan di lakukan perhitungan manual dari beberapa titik menggunakan metode haversine untuk membuktikan bahwa rumus haversine dapat menghitung jarak antara dua buah titik di permukaan bumi. Berikut adalah perhitungan manual metode haversine dengan menghitung jarak Antara user dan BPBD berdasarkan latitude dan longitude.

$$R = 6371$$

$$\theta_1 = \text{latitude user}$$

$$\theta_2 = \text{latitude BPBD}$$

$$\lambda_1 = \text{longitude user}$$

$$\lambda_2 = \text{longitude BPBD}$$

contoh perhitungan :

lokasi user Cebongan Kidul, Tlogoadi Kec. Mlati Kabupaten Sleman Daerah Istimewa Yogyakarta 55286

$$\theta_1 : -7.696338$$

$$\lambda_1 : 110.394686$$

#### a. Perhitungan 1

BPBD Jl. Ipda Tut Harsono No.8, Muja Muju, Kec. Umbulharjo, Kota Yogyakarta 55165

$$\theta_2 : -7.8010158$$

$$\lambda_2 : 110.3918219$$

$$\Delta lat = \frac{\pi}{180} * (\theta_2 - \theta_1) = \frac{3,14}{180} * (-7.8010158 - (-7.696338)) = -0.00183$$

$$\Delta long = \frac{\pi}{180} * (\lambda_2 - \lambda_1) = \frac{3,14}{180} * (110.3918219 - 110.394686) = 0.0000498898$$

$$a = \sin\left(\frac{\Delta lat}{2}\right) = \sin^2\left(\frac{-0.00329}{2}\right) = 0.00000083722$$

$$c = \cos(\theta_1) * \cos(\theta_2) * \sin^2\left(\frac{\Delta long}{2}\right) = \cos(-7.696338) * \cos(-7.8010158) * \sin^2\left(\frac{0.0000498898}{2}\right) = 0.000000003.4792$$

$$d = R * 2 * \arcsin\left(\sqrt{a + c}\right) = \frac{6371}{2} * \arcsin\left(\sqrt{0.00000083722 + 0.000000003.4792}\right)$$

$$= 9,74 \text{ km}$$

#### b. Perhitungan 2

Jl. Kenari No.14A, Semaki, Kec. Umbulharjo, Kota Yogyakarta 55166

$$\text{lat}_2 : -7.798547$$

long2 : 110.3869516

$$\Delta lat = \frac{\pi}{180} * (\theta_2 - \theta_1) = \frac{3,14}{180} * (-7.798547 - (-7.696338))$$
$$= -0.001782$$

$$\Delta long = \frac{\pi}{180} * (\lambda_2 - \lambda_1) = \frac{3,14}{180} * (110.3869516 - 110.394686)$$
$$= -0.00013492$$

$$a = \sin\left(\frac{\Delta lat}{2}\right) = \sin^2\left(\frac{-0.001782}{2}\right)$$
$$= 0.000000013871$$

$$c = \cos(\theta_1) * \cos(\theta_2) * \sin^2\left(\frac{\Delta long}{2}\right)$$
$$= \cos(-7.696338) * \cos(-7.798547) * \sin^2\left(\frac{-0.00013492}{2}\right)$$
$$= -0.00000000118848$$

$$d = R * 2 * \text{asin}\left(\frac{\sqrt{a^2 + c}}{2}\right)$$
$$= \frac{6371 * 2 * \text{asin}\left(\frac{\sqrt{0.000000013871 + (-0.00000000118848)}}{2}\right)}{1}$$
$$= 9.31\text{km}$$

### 5.1.1 Spesifikasi perangkat keras

Dalam menerapkan dari rancangan yang telah dijelaskan sebelumnya dibutuhkan beberapa perangkat keras untuk menyajikan aplikasi ini. Adapun alat-alat yang dibutuhkan adalah :

1. Sistem operasi : Android 6.0 (Marshmallow)

Handphone berbasis sistem operasi Android digunakan untuk menjalankan program aplikasi yang telah dikembangkan. Adapun handphone yang digunakan adalah Samsung S5 dengan spesifikasi sebagai berikut :

- a. CPU : Qualcomm Snapdragon 801
- b. Ruang Penyimpanan : 16 GB
- c. Memory : 2 GB
- d. Dimensi Layar : 5.1 inche

1. Kabel data serial port

Fungsi dari kabel data ini adalah untuk menghubungkan antara Laptop dengan Handphone.

2. Satu unit laptop dengan spesifikasi

- a. CPU : Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz (4 CPUs), ~2.6GHz
- b. Memori : 6 GB
- c. Ruang penyimpanan : 500 GB

### 5.1.2 Spesifikasi Perangkat Lunak

1. Bahasa pemrograman Java

Dalam hal ini digunakan bahasa pemrograman *Java Development Kit (JDK)* dan *Java Runtime Environment (JRE)*

2. Sistem Operasi

Sistem operasi yang digunakan adalah Windows 10 pro 64 bit

3. Android Studio 3.4.2

Untuk memudahkan dalam pengembangan aplikasi ini maka digunakan Android Studio karena memiliki beberapa fasilitas yang diperlukan dalam pengembangan aplikasi Android.

4. Android Software Development Kit (SDK)

Android SDK menyediakan *development environment* dengan semua komponen yang diperlukan. Antara lain *tools* pengembangan, *libraries*, dokumentasi, serta disediakan pula emulator untuk mensimulasi aplikasi berjalan pada perangkat.

### 5.1.3 Implementasi Antarmuka Sistem

Berikut ini adalah implementasi antarmuka *Implementasi Metode Haversine Formula dalam Aplikasi untuk Menentukan Lokasi Emergency Service Terdekat di Daerah Istimewa Yogyakarta* berbasis Android yang terdiri dari tampilan Android.

#### 5.1.3.1 Halaman *Splashscreen Emergency Service*

*Splashscreen* ini adalah menu yang akan ditampilkan pertama kali ketika pengguna/user mengakses aplikasi. *Splashscreen* aplikasi *emergency\_service* dapat dilihat pada Gambar 5.1.



Gambar 5.1 Halaman *Splashscreen*

#### 5.1.3.2 Halaman Menu Utama *Emergency Service*

Halaman menu utama pada aplikasi *emergency service* ini menampilkan menu BPBD, menu Rumah Sakit, menu Kantor Polisi, menu Berita, dan menu tentang. Halaman menu aplikasi *emergency service* dapat dilihat pada Gambar 5.2.



Gambar 5.2 Halaman Menu Utama

#### 5.1.3.3 Halaman Menu BPBD

Halaman menu BPBD ini menampilkan semua alamat pos-pos BPBD yang ada di Daerah Istimewa Yogyakarta. Halaman menu BPBD dapat dilihat pada Gambar 5.3.



### Gambar 5.3 Halaman Menu BPBD

#### 5.1.3.4 Halaman Menu Polisi

Halaman menu Polisi ini menampilkan semua alamat kantor Polisi yang ada di Daerah Istimewa Yogyakarta. Halaman menu Polisi dapat dilihat pada Gambar 5.8.



Gambar 5.8 Halaman Menu Polisi

#### 5.1.3.5 Halaman Menu Kesehatan

Halaman menu Kesehatan ini menampilkan semua alamat Kesehatan yang ada di Daerah Istimewa Yogyakarta. Halaman menu Kesehatan dapat dilihat pada Gambar 5.13.



Gambar 5.13 Halaman Menu Kesehatan

#### 5.1.3.6 Halaman Menu News

Halaman News ini menampilkan berita dan informasi yang ter-update di Yogyakarta. Halaman news dapat dilihat pada gambar 5.18.



Gambar 5.18 Halaman News

#### 5.1.3.7 Halaman Menu Tentang

Halaman menu tentang berisi informasi tentang pembuat aplikasi. Halaman Menu tentang dapat dilihat pada Gambar 5.19.



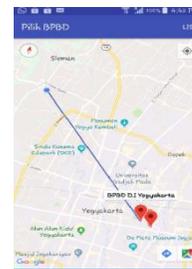
Gambar 5.19 Halaman Menu Tentang

### 5.2 Implementasi Metode Haversin

Berikut adalah tabel hasil perhitungan Haversine dan gambar jarak user dengan titik emergency service kota Yogyakarta, pada uji coba ini lokasi user berada di Cebongan Kidul, Tlogoadi Kec. Mlati Kabupaten Sleman Daerah Istimewa Yogyakarta 55286 dengan latitude ( $\theta_1$ ) = -7.696338 dan longitude ( $\lambda_1$ ) = 110.394686

#### 5.2.1 Implementasi Metode Pada Menu BPBD

Berikut adalah tabel hasil perhitungan jarak BPBD dengan User dan implementasi metode Haversine pada menu BPBD dapat di lihat pada gambar 5.20.



Gambar 5.20 Halaman menu BPBD

#### 5.2.2 Implementasi pada Menu Polisi

Berikut adalah implementasi metode Haversine pada menu Polisi dapat di lihat pada gambar. 5.21.



#### 5.2.3 Implementasi Metode Dan Perhitungan Pada Menu Kesehatan

Berikut adalah tabel hasil perhitungan jarak Kantor Polisi dengan User dan implementasi metode Haversine pada menu kesehatan dapat di lihat pada gambar 5.22.



Gambar 5.22 Halaman menu kesehatan

### 5.3 Hasil Pengujian Sistem

*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya atau antarmuka, fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya atau hanya mengetahui *input* dan *output*. dengan demikian penulis akan melakukan pengujian sistem menggunakan *black box* kemungkinan pembuatan perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Hasil dari pengujian sistem ini dapat dilihat Tabel 5.1.

Tabel 5.4 Hasil Pengujian Menggunakan *Black Box*

Tes Case	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
GPS belum aktif.	Masuk ke dalam aplikasi <i>emergency service</i> .	Aplikasi akan meminta untuk masuk ke pengaturan untuk mengaktifkan GPS.	Keluar permintaan untuk masuk ke pengaturan ke GPS.	<i>valid</i>
Masuk ke menu BPBD, me	Paket data tidak aktif.	Aplikasi akan memberikan info untuk cek paket internet.	Muncul info gagal koneksi ke server, periksa koneksi internet anda!	<i>valid</i>

nu Kesehatan, dan Polisi.				
Masuk ke menu BPBD, menu Kesehatan dan menu Kantor Polisi.	Paket data aktif.	Aplikasi akan menampilkan list daftar alamat BPBD, daftar Kesehatan dan daftar alamat Kantor Polisi.	Muncul daftar <i>list</i> alamat BPBD, daftar Kesehatan dan daftar Kantor Polisi.	<i>valid</i>
Masuk ke menu Nws.	Paket data tidak aktif.	Muncul <i>error</i> .	<i>Internet_Disonected</i>	<i>Valid</i>
Masuk ke menu Nws.	Paket data aktif.	Muncul <i>website</i> : <a href="http://www.jogjakota.go.id">http://www.jogjakota.go.id</a> .	Tampil halaman situs web <a href="http://www.jogjakota.go.id">http://www.jogjakota.go.id</a>	<i>alid</i>

## 6. PENUTUP

### 6.1 Kesimpulan

Berdasarkan pembahasan dari bab-bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi ini dibangun untuk memudahkan pengguna pada saat keadaan darurat yaitu masyarakat, dalam mencari informasi mengenai lokasi *emergency service*.
2. Aplikasi *emergency service* telah berhasil diimplementasikan menggunakan fasilitas Google *Maps* pada perangkat Android yang telah dilengkapi GPS.

3. Aplikasi yang dibangun dapat mencakup *emergency service* di Daerah Yogyakarta.

## 6.2 Saran

Perancangan dan implementasi yang telah dilakukan ini masih jauh dari sempurna, untuk penelitian selanjutnya terdapat beberapa saran yang dapat dipakai untuk pengembangan yang lebih baik lagi.

1. Diharapkan sistem ini mampu menjangkau atau mendeteksi di mana *user* berada, meskipun *user* berada di wilayah terpencil sekalipun.
2. Mengembangkan aplikasi *emergency service* sehingga dapat dijalankan pada sistem operasi yang lain (*multi platform*).

## Daftar Pustaka

- [1] Yulianto, B and Layona, R, (2016), *Aplikasi Pencarian Tempat Wisata Berbasis GPS dengan Metode Radius dan Rating*, Skripsi, Binus University.
- [2] Satria, G. and Febriani, F. F (2017) *Pemanfaatan Location Based Service Berbasis Android untuk Pemetaan Kantor Polisi Wilayah Kota Pekanbaru*, Jurnal Penelitian Sains, FMIPA Universitas Riau
- [3] Yulianto, W., (2015), *Menentukan Jarak Terdekat Hotel dengan Metode Haversine Formula*, Skripsi, Universitas Islam Negeri Maulana Malik Ibrahim Malang
- [4] Usman, N., (2002), *Konteks Implementasi Berbasis Kurikulum*, PT. Raja Grafindo Persada, Jakarta.
- [5] Jogiyanto H., (2004), *Analisis dan Desain*, Andi Offset, Yogyakarta.
- [6] Soegimo, D. dan Ruswanto., (2009), *Geografi untuk SMA/MA Kelas X*, Pusat Perbukuan, Jakarta.
- [7] Supriadi, Y., (2014), *Senua Bisa Menjadi Programmer Android Case Study*, PT. Elex Medis Komputindo, Jakarta.
- [8] Ariyanti, R and khairi, I.K (2015), *Memfaatkan Google Maps API pada Sistem Informasi Geografis Direktori Perguruan Tinggi di Kota Bengkulu*, Jurnal Medis Infotama, Universitas Dehasen Bengkulu.
- [9] Alfena, S, and Eka, R.D., (2017), *Implementasi Global Positioning System (GPS) dan Location Based Service (LBS) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek*, Jurnal Sisfotek Global, STMIK Raharja.
- [10] Juansyah, A., (2015), *Pembangunan Aplikasi Child Tracker Berbasis Assisted - Global Positioning System (A-GPS) dengan Platform Android*, Jurnal Ilmiah Komputer dan Informatika, Universitas Komputer Indonesia.
- [11] Silberschatz, dkk. (2011), *Database System Concepts Edisi 6*, McGrawHill Companies, New York.



