

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tabel 2.1 Tinjauan Pustaka

Penulis	Objek	Metode	Bahasa Pemrograman	Interface
Febri Nugroho (2016)	Kantor Polisi	Location Based Service	Java	Android
Anugerah Panembahan Attunggal (2016)	Bengkel Motor	Location Based Service	Java	Android
Ismail (2013)	Perguruan Tinggi	Location Based Service	Java	Android
Muhammad Aji Pratama (2013)	SMA & SMK Negeri	Location Based Service	Java	Android
Arris Budi Yatno (2015)	Apotek	Location Based Service	Java	Android
Nur Isnaini Agustina (2017)	Perguruan Tinggi	Location Based Service	Java	Android

Dalam ulasan penelitian ini yang membedakan antara usulan dengan pustaka yang sudah ada adalah terdapat menu pencarian dengan menggunakan kata kunci ,serta aplikasi akan menampilkan gambar objek lokasi agar pengguna mengetahui sekilas tentang lokasi melalui gambar.

2.2 Dasar Teori

2.2.1 Sistem Informasi Geografis (GIS)

Menurut ESRI (*Environmental System Research Institute*) mendefinisikan SIG adalah suatu komponen yang terdiri dari perangkat lunak, perangkat keras, data geografis dan sumber daya manusia yang bekerja sama secara efektif untuk menangkap, menyimpan, memperbaiki, memperbarui, mengelola, memanipulasi, mengintegrasikan, menganalisa dan menampilkan data dalam suatu informasi berbasis geografis.

SIG merupakan suatu sistem informasi yang dapat memadukan antara data grafis dengan data teks (*atribut*) objek yang dihubungkan secara geografis di bumi (*georeference*). Di samping itu, sistem Informasi Geografis ini juga dapat menggabungkan data mengatur data dan menganalisis data, untuk selanjutnya menghasilkan output yang dapat dijadikan acuan dalam pengambilan keputusan pada masalah geografis. (Riyanto dan Hendi Inderaloka, 2009).

2.2.2 Google Maps

Google maps merupakan sebuah layanan peta dunia virtual berbasis web yang disediakan oleh google. Layanan ini gratis dan dapat ditemukan di <http://maps.google.com>. Google maps menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*) dapat diganti dalam beberapa mode (map, satellite, hybrid, dan lain-lain), fitur pencarian rute (*direction*) dan juga pencarian tempat (*place*). (Riyanto, 2010)

2.2.3 Android

Android adalah system operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. (Nazarudin Safaat ,2012).

2.2.4 *Location Based Service (LBS)*

Location Based Service (LBS) adalah sebuah layanan untuk menggambarkan teknologi yang dapat menemukan suatu lokasi.

Dua unsur utama *Location Based Service (LBS)* :

1. Location Manager (API Maps)

Menyediakan *tools* atau *source* untuk LBS, *Application Programming Interface (API Maps)* menyediakan fasilitas menampilkan memanipulasi *maps* atau peta beserta *feature-feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya.

2. Location Providers (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device* (perangkat). *API location* berhubungan dengan data *Global Positioning System (GPS)* dan data lokasi *real-time*. GPS merupakan sistem koordinat global yang dapat menentukan koordinat posisi benda dimana saja di bumi baik koordinat lintang dan bujur (*latitude* dan *longitude*), maupun ketinggiannya. *API location* berada pada paket android yaitu dalam paket *android.location*. dengan *Location*

Manager, penulis dapat menentukan lokasi saat itu juga. (*real-time*). (Riyanto, 2010)

2.2.5 Global Positioning System (GPS)

Global Positioning System(GPS) adalah sistem navigasi dan penentuan posisi menggunakan satelit yang dikembangkan dan dikelola oleh Departemen Pertahanan Amerika Serikat. GPS dapat memberikan informasi tentang posisi, kecepatan dan waktu di mana saja di muka bumi setiap saat, dengan ketelitian penentuan posisi dalam fraksi mili meter sampai dengan meter. Kemampuan jangkauannya mencakup seluruh dunia dan dapat digunakan banyak orang setiap saat pada waktu yang sama. (Abidin, 2007).

2.2.6 JSON (Java Script Object Notation)

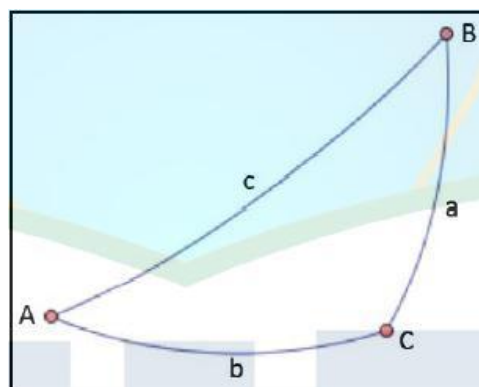
JSON (*JavaScript Object Notation*) adalah format pertukaran data (*lightweight data interchange format*) , mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman Java Script, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java ,Java Script, Perl, Python dll. Oleh karena itu sifat-sifat tersebut menjadikan JSON ideal sebagai bahasa pertukaran data.(Nazaruddin Safaat,2012)

2.2.7 *Latitude dan Longitude*

Latitude disebut juga garis lintang .Garis lintang merupakan garis vertical yang mengukur sudut antara suatu titik dengan garis katulistiwa.Titik di utara garis katulistiwa dinamakan lintang utara, sedangkan titik diselatan katulistiwa dinamakan lintang selatan. *Longitude* disebut juga garis bujur. Garis bujur yaitu garis horizontal yang mengukur sudut antara satu titik dengan titik nol bumi, yaitu Greenwich di London,Britania Raya yang merupakan titik 0^0 atau 360^0 yang diterima secara internasional.Titik dibarat bujur 0^0 dinamakan bujur barat,sebaliknya titik di timur 360^0 dinamakan bujur timur. (Sirenden dan Dachi,2012).

2.2.8 *Spherical Law of Cosines*

Spherical Law of Cosines merupakan salah satu persamaan dasar *spherical triangle*. Salah satu pengaplikasian dari *Spherical Law of Cosines* adalah mengkalkulasi jarak diantara dua titik diatas permukaan bumi. Untuk mengetahui bagaimana *Spherical Law of Cosines* digunakan, perhatikan gambar dibawah ini. (Benerjee,2004) .



Gambar 2.1 Spherical triangle

Gambar 2.1 merupakan spherical triangle dengan titik A,B,C dan sisi melengkung a, b dan c. Sisi melengkung tersebut merupakan jarak geodetik yang bisa diketahui jaraknya. Apabila $(lat1, long1)$ dan $(lat2, long2)$ merupakan koordinat geografis dari titik B dan C, maka bisa didapat nilai $b = -\theta_1$, $c = -\theta_2$, dan $A = 2 - 1$. Kemudian untuk mencari jarak antar B dan C dengan menggunakan rumus *Spherical Law of Cosines*, ekspresinya adalah sebagai berikut : p

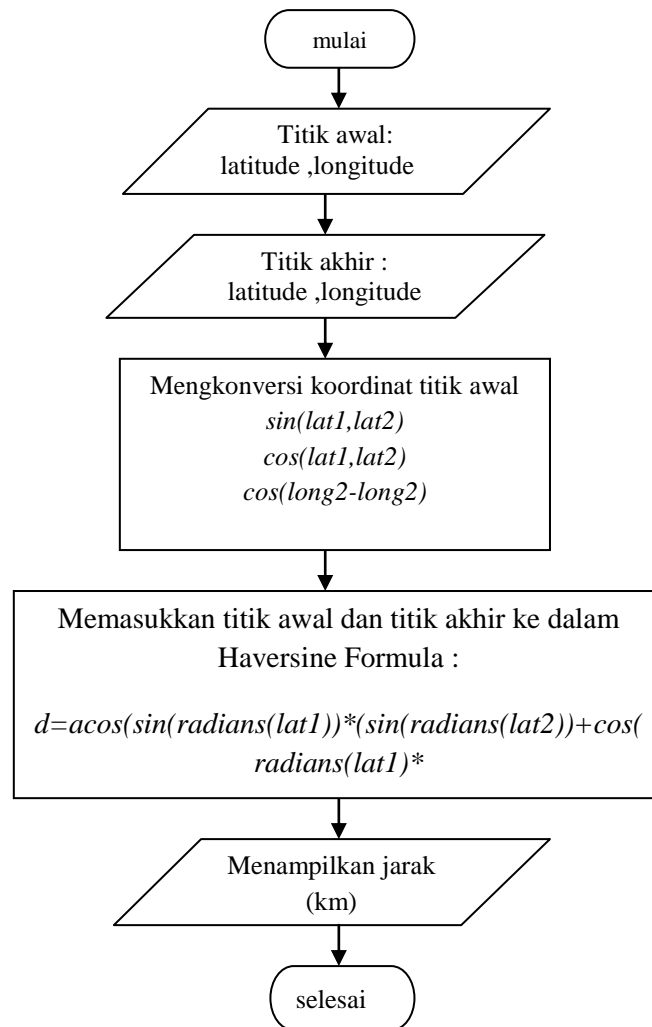
$$d = \text{acos}(\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2\cos A).R \quad (1)$$

$$d = \text{acos}(\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2\cos(2 - 1)).R \quad (2)$$

$$d = \text{acos}(\sin(lat1).\sin(lat2) + \cos(lat1).\cos(lat2).\cos(long2-long1)).R \quad (3)$$

Keterangan :

- d adalah jarak antar dua point.
- Lat adalah latitude.
- $long$ adalah longitude.
- R adalah radius dari lingkaran bola ($R = 637.100$: radius Bumi dalam meter). (Nurul,2015).



Gambar 2.2 Flowchart

Gambar 2.2 merupakan flowchart menghitung Haversine Formula yang secara otomatis sistem akan mendeteksi koordinat *latitude longitude* titik awal posisi pengguna dan *latitude longitude* titik akhir tujuan pengguna. Kemudian mengkonversi koordinat titik awal dan titik akhir ke dalam radian dan selanjutnya dilakukan pencarian jarak antara posisi awal dengan posisi akhir menggunakan Haversine Formula dan mengkonversi hasilnya ke dalam kilometer (Km) dengan mengalikan dengan $R=6371$.