

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan Pustaka menguraikan hasil-hasil karya ilmiah yang pernah dilakukan oleh penulis yang karyanya mempunyai kaitan dengan proyek akhir yang dilakukan. Pada bagian ini membandingkan dan mereferensi solusi masalah kita dengan solusi lain melalui studi literatur.

Tabel 2. 1 Perbedaan dan Tinjauan Pustaka

parameter Penulis	Objek Penelitian	Metode/ Teknologi	Database	Interface	Hasil
Sendi Prasojo (2016)	Aplikasi Pencarian Tempat Wisata	OSM Real Time Rute	MySQL	Android	Menampilkan informasi tempat wisata di kabupaten Kebumen
Gunawan, Arif (2016)	Aplikasi pencarian Lokasi Kantor Polisi Terdekat	Location Based Service	MySQL	Android	Menampilkan rute menuju kantor polisi terdekat di wilayah Bantul
Trias Retno Wulan Suci (2017)	Aplikasi Pencarian Lokasi Rumah Sakit Dan Jadwal Praktek Doktor	Location Based Service	MySQL	Android	Menampilkan rute rumah sakit terdekat

parameter Penulis	Objek Penelitian	Metode/ Teknologi	Database	Interface	Hasil
Nur Isnaini (2016)	Sistem Informasi Pencarian Perguruan Tinggi Di Yogyakarta	Location Based Service	MySQL	Android	Menampilkan rute dan informasi perguruan tinggi yang berada di yogyakarta
Fransiscus Dato Key	Aplikasi Pencarian Halte Trans jogja	Haversine	MySQL	Android	Menampilkan informasi halte terdekat berdasarkan tujuan pengguna.

Dalam membuat aplikasi ini merujuk dari empat penelitian yang sudah ada. Pada tahun 2016 sebelumnya pernah dilakukan penelitian tentang pencarian tempat wisata, dengan judul “ Aplikasi Pencarian Tempat Wisata Di Kabupaten Kebumen Dengan Mengguankan OSM Real Time Rute berbasis Android”, yang dibuat oleh mahasiswa STMIK Akakom bernama Sendi Prasajo dengan NIM 105410134. Aplikasi ini dapat menampilkan tempat wisata dan mengetahui *longitude* dan *latitude* pengguna.

Pada tahun 2016 pernah dibuat “Aplikasi pencarian Lokasi Kantor Polisi Terdekat Di Kabupaten Bantul Dengan Metode Location Based Service Berbasis Android “ oleh mahasiswa STMIK Akakom yang bernama Febri Nugroho dengan NIM 125410243. Aplikasi ini dapat mencari lokasi kantor polisi terdekat dari posisi pengguna.

Pada tahun 2017 pernah dibuat “Aplikasi Pencarian Lokasi Rumah Sakit Dan Jadwal Praktek Dokter Berbasis Android”, oleh mahasiswi STMIK Akakom bernama Trias Retno

Wulan Suci dengan NIM 135410227. Aplikasi ini menampilkan lokasi rumah sakit terdekat disertai dengan jam praktik Dokter.

Pada tahun 2016 pernah dibuat “Sistem Informasi Pencarian Perguruan Tinggi Di Yogyakarta Berbasis Android” aplikasi ini dibuat oleh mahasiswi STMIK Akakom yang bernama Nur Isnaini Agustina dengan NIM 125410242. Aplikasi ini digunakan untuk memberikan informasi perguruan tinggi di Yogyakarta yang dapat dibedakan berdasarkan jenis perguruan tinggi.

Berdasarkan keempat rujukan diatas penulis berencana membuat sebuah aplikasi “Pencarian Halte Transjogja Berbasis Android Menggunakan Formula Haversine”, aplikasi ini digunakan untuk menemukan halte terdekat serta dapat menuntun pengguna menuju halte yang telah ditentukan. Pada kali ini yang menjadi pembeda pada aplikasi pencarian yang sebelumnya adalah penggunaan metode yaitu Metode Haversine serta objek pencarian adalah halte trans jogja.

2.2 Dasar Teori

2.2.1 Android

Android adalah [sistem operasi](#) berbasis [Linux](#) yang dirancang untuk perangkat bergerak [layar sentuh](#) seperti [telepon pintar](#) dan [komputer tablet](#). Android awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari [Google](#), yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya [Open Handset Alliance](#), *konsorsium* dari perusahaan-

perusahaan [perangkat keras](#), perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan [standar terbuka](#) perangkat seluler. (Andry, 2011)

2.2.2 GPS

Global Positioning System (GPS) adalah sistem radio navigasi dan penentuan posisi menggunakan satelit, dengan nama resminya NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System*). GPS dikembangkan pertama kali oleh Departemen Pertahanan Amerika Serikat pada tahun 1978 dan secara resmi GPS dinyatakan operasional pada tahun 1994. Pada awalnya GPS digunakan hanya untuk kepentingan militer, tetapi kemudian dapat dimanfaatkan juga untuk kepentingan sipil. (Abidin,H.Z, 2007)

2.2.3 Google API (*application programming interfaces*)

Google menyediakan berbagai *API* yang sangat berguna bagi pengembang aplikasi. Berbagai macam *Api* telah disediakan oleh *Google*. *API* secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi yang kita buat dengan fungsi-fungsi yang dikerjakan. *Google api* menyediakan berbagai utilitas yang dapat memanipulasi peta dan menambahkan konten didalam peta. Sebagai contoh terdapat *Direction API* yang berfungsi untuk menghitung arah antar lokasi, serta bisa menelusuri arah untuk beberapa moda transportasi, termasuk angkutan umum, mengemudi, berjalan atau bersepeda.

2.2.4 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. *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*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data.

2.2.5 PHP (*Hypertext Preprocessor*)

PHP adalah *script* bersifat *server side* yang ditambahkan ke dalam *HTML*. Kelebihan PHP yang paling signifikan adalah kemungkinan untuk melakukan koneksi dengan berbagai macam database. PHP merupakan Bahasa *interpreter* yang memiliki kesederhaan dalam perintah. PHP dapat digunakan untuk *mengupdate database*, menciptakan *database* dan mengerjakan perhitungan matematika

PHP adalah Bahasa (*Script language*) yang dirancang secara khusus untuk penggunaan Bahasa *web*. PHP adalah *tool* untuk pembuatan halaman web dinamis seperti bahasa pemrograman *web* lainnya. PHP memproses seluruh perintah yang berada dalam *script* PHP di dalam *web server* dan menampilkan outputnya ke dalam *web browser client*.

(Arief, M.Rudianto. 2011)

2.2.6 MySql

MySql adalah multiuser database yang menggunakan bahasa *SQL (Structured Query Language)*. *SQL* adalah bahasa standar yang digunakan untuk mengakses *database server*.

2.2.7 Metode Haversine

Metode *Haversine* digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variabel inputan. *Haversine formula* adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. (Dwi Prasetyo dkk, 2014)

Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon_1 , lat_1 , dan lon_2 , lat_2 , maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut :

$$\begin{aligned}x &= (lon_2 - lon_1) * \cos((lat_1 + lat_2)/2); \\y &= (lat_2 - lat_1); \\d &= \sqrt{x^2 + y^2} * R\end{aligned}$$

Keterangan:

x = Longitude (Lintang)

y = Latitude (Bujur)

d= Jarak

R= Radius Bumi =6371 km

1 derajat= 0.0174532925 radian

Contoh Perhitungan Haversine Formula

Lokasi 1: lon1= 119.800801, lat1= -0.790175

Lokasi 2: lon2= 119.8428 , lat2= -0.8989

lat1 = -0.790175 * 0.0174532925 radian = -0.013791155 radian

lon1= 119.800801 * 0.0174532925 radian = 2.090918422 radian

lat2 = -0.8989 * 0.0174532925 radian = -0.01569 radian

lon2= 119.8428 * 0.0174532925 radian= 2.091651 radian

$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2}) / 2)$

$= (2.091651 - 2.090918422) * \cos ((-0.013791155 + -0.01569) / 2)$

$= 0.0007329412$

$y = (\text{lat2} - \text{lat1})$

$= (-0.01569 - (-0.013791155))$

$= -0.001897609$

$d = \text{sqrt} (x * x + y * y) * R$

$= \text{sqrt}((0.0007329412 * 0.0007329412) + (-0.001897609 * -$

$0.001897609)) * 6371$

$= \text{sqrt} (0.0000041381) * 6371$

=12.96012927 km