

APLIKASI PENGHITUNG JARAK KOORDINAT BERDASARKAN LATITUDE DAN LONGITUDE DENGAN METODE EUCLIDEAN DISTANCE DAN METODE HAVERSINE

Canggih Ajika Pamungkas

Politeknik Indonusa Surakarta
Jl. KH. Samanhudi No. 31 Mangkuyudan Surakarta
Email : canggih@poltekindonusa.ac.id

Abstrak

Pada era globalisasi, segala aspek kehidupan manusia dituntut untuk semakin maju dan berkembang dengan cara memanfaatkan teknologi informasi. Saat ini teknologi informasi merupakan salah satu kebutuhan primer bagi masyarakat modern. Teknologi informasi telah mempengaruhi kehidupan masyarakat modern. Salah satu teknologi informasi yang dimanfaatkan masyarakat adalah Google Maps. Google Maps merupakan pelayanan pemetaan web yang dikembangkan oleh Google. Google Maps memberikan pelayanan citra satelit. Fitur Google Maps dapat disematkan pada sebuah website dengan Google Maps API. Google Maps API merupakan library JavaScript.

Artikel ini menyajikan pembahasan mengenai pembangunan Aplikasi Berbasis Android Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude menggunakan metode *Euclidean Distance* dan metode *Haversine* memanfaatkan API Google Maps. Pengembangan yang dilakukan menggunakan HTML, PHP, CSS dan Javascript. Aplikasi web tersebut kemudian dibentuk menjadi file *.apk dengan *software* Website 2 APK Builder.

Hasil dari penelitian menunjukkan aplikasi dapat digunakan pada perangkat android. Saat pengujian menunjukkan hasil sama saat perhitungan jarak antara metode *Euclidean Distance* dan metode *Haversine*.

Kata Kunci : *Euclidean Distance, Haversine, Google Maps API, Android*

1. PENDAHULUAN

Pada era globalisasi seperti saat ini, segala aspek kehidupan manusia dituntut untuk semakin maju dan berkembang dengan cara memanfaatkan teknologi informasi. Saat ini teknologi informasi merupakan salah satu kebutuhan primer bagi masyarakat modern. Teknologi informasi telah mempengaruhi kehidupan masyarakat modern. Teknologi informasi yang semakin berkembang memberikan pengaruh untuk mengubah pola kerja agar lebih menyesuaikan dengan perkembangan teknologi yang ada. Dampak ketergantungan manusia terhadap teknologi informasi mampu meningkatkan produktivitas kerja. Salah satu teknologi informasi yang dimanfaatkan masyarakat adalah Google Maps.

Google Maps merupakan pelayanan pemetaan web yang dikembangkan oleh Google. Google Maps memberikan pelayanan citra satelit. Fitur Google Maps dapat disematkan pada sebuah website dengan Google Maps API. Google Maps API

merupakan library JavaScript. Google Map ternyata dapat dikembangkan menjadi aplikasi yang mampu menyajikan informasi mengenai kondisi geografis. Artikel ini menyajikan pembahasan mengenai pembangunan Aplikasi Berbasis Android Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude menggunakan metode *Euclidean Distance* dan metode *Haversine* memanfaatkan API Google Maps. Pengembangan yang dilakukan menggunakan HTML, PHP, CSS dan Javascript. Aplikasi web tersebut kemudian dibentuk menjadi file *.apk dengan *software* Website 2 APK Builder.

2. METODE PENELITIAN

a. Metode Pengumpulan Data

1) Studi Pustaka

Studi pustaka merupakan teknik pengumpulan data dengan mengadakan studi penelaahan terhadap buku-buku, literatur-literatur, catatan-catatan, dan laporan-laporan

yang ada hubungannya dengan masalah yang dipecahkan. Dalam penelitian ini penulis melakukan studi pustaka dengan mengambil referensi buku-buku yang membahas metode *Euclidean Distance* dan metode *Haversine*.

b. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah waterfall. Berikut Fase-fase dalam Waterfall Model:



Gambar 1. Metode Waterfall

3. TINJAUAN PUSTAKA

a. Penelitian Terdahulu

Sariyun dkk (2015) melakukan penelitian dengan judul Model Rute dan Peta Interaktif Posyandu di Kota Semarang menggunakan Geolocation dan *Haversine* Berbasis Mobile Android. Hasil penelitian ini ialah terbentuknya aplikasi rute dan peta navigasi berbasis mobile Android yang dapat memberikan informasi mengenai keberadaan lokasi, rute dan jarak posyandu. Hasil aplikasi ini penting bagi masyarakat yang membutuhkan petunjuk jalan lokasi posyandu. Aplikasi ini mampu membantu masyarakat memperoleh informasi rute dan lokasi posyandu di kota Semarang.

Yulianto dkk (2018) melakukan penelitian dengan judul Penerapan Formula *Haversine* Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal. Berdasarkan analisis dan pengujian yang dilakukan, maka dapat ditarik kesimpulan bahwa sistem mampu menghasilkan pemetaan lokasi lapangan futsal di kota Samarinda dilengkapi dengan informasi mengenai lokasi tersebut. Selain itu dapat ditarik kesimpulan bahwa penerapan formula *Haversine* mampu memberikan informasi jarak dari lokasi pengguna ke lokasi lapangan futsal. Kesimpulan lainnya adalah Formula *Haversine* dapat dimanfaatkan sebagai pencarian jarak terdekat dengan cara mencari hasil yang paling kecil nilainya sebagai lokasi dengan jarak terdekat.

b. Latitude

Latitude atau garis lintang merupakan garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator. Garis lintang diukur mulai dari titik 0 derajat dari khatulistiwa sampai 90 derajat di kutub.

c. Longitude

Longitude atau garis bujur merupakan digunakan untuk menentukan lokasi di wilayah barat atau timur dari garis utara selatan yang sering disebut juga garis meridian. Garis bujur diukur dari 0 derajat di wilayah Greenwich sampai 180 derajat di International Date Line.

d. Euclidean

Formula *euclid* merupakan salah satu formula yang digunakan untuk mengukur jarak dari 2 titik dengan menggunakan perhitungan matematis (metode heuristik).

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 Phytagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

Metode *Euclidean Distance* merupakan suatu metode pencarian kedekatan jarak dari 2 buah variabel, selain mudah metode ini juga lebih efisien waktu, dan proses yang cepat. *Euclidean Distance* merupakan fungsi *heuristik* yang diperoleh berdasarkan jarak langsung bebas hambatan seperti untuk mendapatkan nilai dari panjang garis diagonal pada segitiga. Tetapi sebelum mendapatkan hasil kedua titik harus direpresentasikan ke dalam koordinat 2 dimensi (x, y). Dua buah titik $p1 = (x1, y1)$ dan $p2 = (x2, y2)$ menjadi persamaan berikut (rumus Euclidian):

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

e. Haversine

Rumus *haversine* adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bumi berdasarkan bujur dan lintang. Rumus metode *haversine* mengabaikan efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga

mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi.

$$\text{Jarak} = 2r \cdot \arcsin \left\{ \sqrt{\sin^2 \left(\frac{\text{Lat}_1 - \text{Lat}_2}{2} \right) + \cos(\text{Lat}_1) \cdot \cos(\text{Lat}_2) \cdot \sin^2 \left(\frac{\text{Long}_1 - \text{Long}_2}{2} \right)} \right\}$$

f. Garis Lintang & Garis Bujur

Garis Lintang (*Latitude*) merupakan garis khayalan yang membelah bumi secara horisontal menjadi dua bagian yaitu utara dan selatan. Garis yang membelah itu adalah garis lintang 0 derajat atau disebut garis khatulistiwa (seperti di Pontianak yang dilewati / paralel dengan garis khatulistiwa) atau garis lintang 0 derajat. Di atas khatulistiwa disebut garis Lintang Utara (LU) dan di bawah disebut garis Lintang Selatan (LS). Garis lintang berkisar dari 0 derajat di khatulistiwa sampai 90 derajat di Kutub Utara atau Kutub Selatan. Besaran tertinggi garis lintang adalah 90 derajat. Sudut lintang, dinotasikan dengan huruf Yunani phi yang diukur dalam derajat, menit dan detik atau derajat desimal, utara atau selatan dari khatulistiwa.

Garis Bujur (*Longitude*) merupakan garis khayalan yang membelah bumi secara vertikal menjadi dua bagian, yaitu bagian Timur dan bagian Barat, serta menghubungkan kutub Utara dan Selatan. Sebagai patokan waktu di dunia adalah garis yang tegak lurus dengan garis khatulistiwa dimana awalnya 0 derajat di Royal Observatory Greenwich, Inggris. Ke arah kanan/ Timur merupakan garis Bujur Timur (BT) sedangkan ke arah kiri/Barat merupakan garis Bujur Barat (BB). Garis bujur ini biasanya dinyatakan dalam derajat, menit dan detik, dan dilambangkan dengan huruf Yunani lambda (λ).

g. Android

Menurut Nazaruddin (2012 : 1) android merupakan sistem operasi untuk perangkat *mobile* dengan berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi yang digunakan oleh bermacam perangkat *mobile*. Android dapat digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi lainnya, seperti Symbian di Nokia, iOS di Apple dan BlackBerry OS.

4. HASIL DAN PEMBAHASAN

a. Analisis

1) Analisis Kebutuhan Non Fungsional

Tahap analisis kebutuhan non fungsional menentukan kebutuhan *hardware* serta *software* sebagai kebutuhan yang harus dipenuhi dalam pengembangan aplikasi. *Hardware* yang dibutuhkan adalah komputer dengan spesifikasi *Intel Pentium Dual Core, Memory 1GB, Hard Disk 500GB, Windows 7*. *Hardware* yang lain yang dibutuhkan adalah handphone dengan sistem operasi Android. *Software* yang dibutuhkan adalah API Google Maps, Website 2 APK Builder, Java Development Kit (JDK). Sedangkan bahasa pemrograman yang digunakan HTML, PHP, CSS dan Javascript.

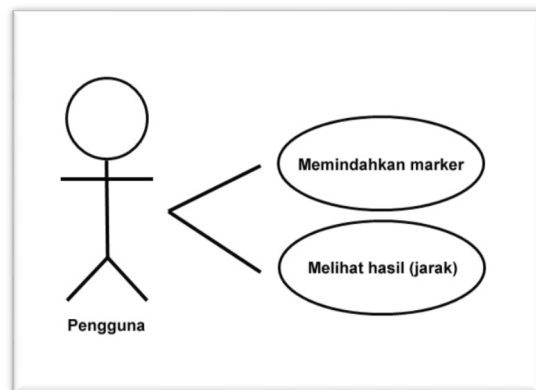
2) Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan langkah pendahuluan sebagai untuk menetapkan tipe program yang akan dikembangkan. Aplikasi yang dikembangkan berbasis android dengan perhitungan berdasarkan *latitude* dan *longitude*. Metode yang digunakan adalah metode *Euclidean Distance* dan metode *Haversine*. Jarak yang dihitung adalah jarak antara dua *marker* yang terdapat pada *map*, dengan status *draggable marker* adalah *true*, sehingga *marker* bisa digeser oleh pengguna.

b. Desain

1) Use Case

Berikut Use case Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude Dengan Metode Euclidean Distance Dan Metode Haversine:

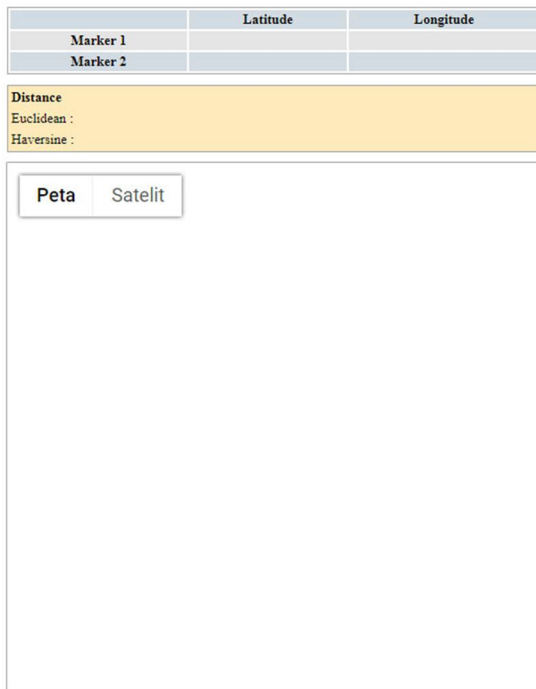


Gambar 2. Use Case Pengguna

2) Rancangan Tampilan

Berikut rancangan tampilan Aplikasi Penghitung Jarak Koordinat Berdasarkan

Latitude dan Longitude Dengan Metode Euclidean Distance dan Metode Haversine:



Gambar 3. Rancangan Tampilan

c. Implementasi

Berikut potongan script Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude Dengan Metode Euclidean Distance dan Metode Haversine:

```
function load() {
    if (GBrowserIsCompatible()) {

        var map = new
        GMap2(document.getElementById("map"));
        map.addControl(new GSmallMapControl());

        map.addControl(new GMapTypeControl());

        var center1 = new GLatLng(-7.79722,
        110.36880);
        var center2 = new GLatLng(-7.79696,
        110.37000);

        map.setCenter(center1, 15);
        map.setCenter(center2, 15);

        var marker1 = new
        GMarker(center1, {draggable: true});
        var marker2 = new
        GMarker(center2, {draggable: true});

        map.addOverlay(marker1);
        map.addOverlay(marker2);

        document.getElementById("latitude1").inn
        erHTML = center1.lat().toFixed(5);
        document.getElementById("longitude1").in
        nerHTML = center1.lng().toFixed(5);

        document.getElementById("latitude2").inn
        erHTML = center2.lat().toFixed(5);
```

```
document.getElementById("longitude2").in
nerHTML = center2.lng().toFixed(5);

function hitung_jarak(){
    var point2 = marker2.getPoint();
    map.panTo(point2);

    document.getElementById("latitude2").inn
erHTML = point2.lat().toFixed(5);

    document.getElementById("longitude2").in
nerHTML = point2.lng().toFixed(5);

    var point1 = marker1.getPoint();
    map.panTo(point1);

    document.getElementById("latitude1").inn
erHTML = point1.lat().toFixed(5);

    document.getElementById("longitude1").in
nerHTML = point1.lng().toFixed(5);

    var
    latitude1=point1.lat().toFixed(5);
    var
    longitude1=point1.lng().toFixed(5);
    var
    latitude2=point2.lat().toFixed(5);
    var
    longitude2=point2.lng().toFixed(5);

    //euclidean
    var distance2 = (Math.sqrt(((latitude2-
    latitude1)*(latitude2-
    latitude1))+((longitude2-
    longitude1)*(longitude2-longitude1)))));
    distance2 = distance2*111.319;
    document.getElementById('euclidean').in
nerHTML = distance2.toFixed(2) ;

    //haversine
    var earth_radius = 6371;
    var dLat = ((latitude2 -
    latitude1)*Math.PI)/180;
    var dLon = ((longitude2 -
    longitude1)*Math.PI)/180;

    var a = Math.sin(dLat/2) *
    Math.sin(dLat/2) +
    Math.cos((latitude1*Math.PI)/180) *
    Math.cos((latitude2*Math.PI)/180) *
    Math.sin(dLon/2) * Math.sin(dLon/2);
    var c = 2 * Math.asin(Math.sqrt(a));
    var d = earth_radius * c;
    document.getElementById('haversine').in
nerHTML = d.toFixed(2) ;

}

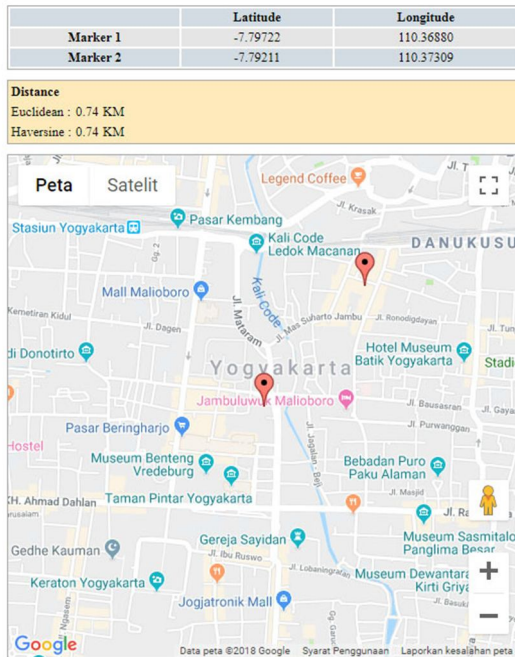
hitung_jarak();

GEvent.addListener(marker2, "dragend",
function() {
    hitung_jarak();
});

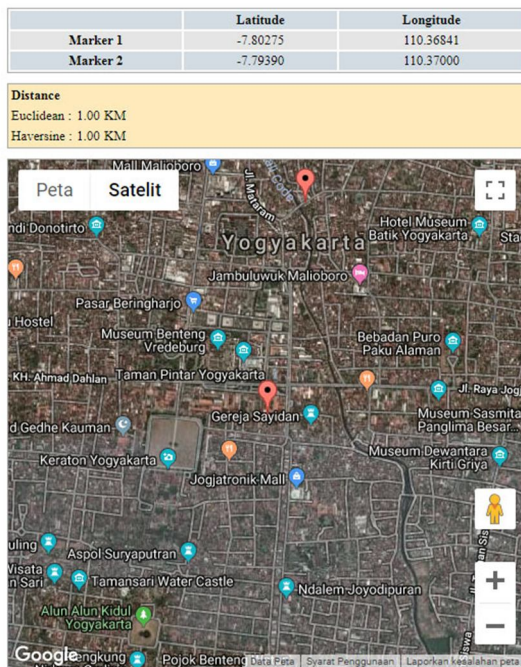
GEvent.addListener(marker1, "dragend",
function() {
    hitung_jarak();
});
}
```

Gambar 4. Script Hitung Jarak

Rumus metode *Euclidean Distance* dan metode *Haversine* diimplementasikan dengan javascript yang dimasukkan dalam function load().



Gambar 5. Tampilan Halaman Aplikasi Tampilan Peta



Gambar 6. Tampilan Halaman Aplikasi Tampilan Satelit

Pada tampilan halaman aplikasi disajikan informasi posisi *marker 1* dan *marker 2*. Selain itu disajikan juga hasil

perhitungan dengan metode *Euclidean Distance* dan metode *Haversine*. Maps disajikan dalam bentuk peta dan satelit sesuai fitur yang disediakan oleh API Google Maps.

d. Pengujian

Tahap pengembangan selanjutnya adalah pengujian aplikasi. Pengujian aplikasi dilakukan untuk mengetahui apakah semua *fitur* pada aplikasi dapat berfungsi dengan baik atau tidak. Dalam pengujian sistem ini, penulis menggunakan suatu metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak yang dibangun.

Metode yang digunakan adalah metode pengujian *Black Box*. Metode pengujian *Black Box* merupakan metode pengujian tanpa memperhatikan struktur logika internal sistem. Metode ini digunakan untuk mengetahui apakah sistem berfungsi sesuai dengan yang diharapkan. Dalam pengujian ini seluruh fitur berhasil menjalankan fungsinya. Berikut hasil pengujian beserta hasil perhitungan rumus metode *Euclidean Distance* dan metode *Haversine*.

Tabel 1. Hasil Pengujian I

	Latitude	Longitude
Marker 1	-7.80275	110.36841
Marker 2	-7.79390	110.37000
Jarak	Metode	
	Euclidean (km)	Haversine (km)
	1.00	1.00

Tabel 2. Hasil Pengujian II

	Latitude	Longitude
Marker 1	-7.80275	110.36841
Marker 2	-7.77940	110.37223
Jarak	Metode	
	Euclidean (km)	Haversine (km)
	2.63	2.63

Tabel 3. Hasil Pengujian III

	Latitude	Longitude
Marker	-7.84811	110.37069

1		
Marker 2	-7.79696	110.37000
Metode		
Jarak	Euclidean (km)	Haversine (km)
	5.69	5.69

Tabel 4. Hasil Pengujian IV

	Latitude	Longitude
Marker 1	-7.86652	110.36876
Marker 2	-7.79696	110.37000
Metode		
Jarak	Euclidean (km)	Haversine (km)
	7.74	7.74

e. Pemeliharaan

Tahap terakhir dari pengembangan aplikasi ini adalah pemeliharaan. Pemeliharaan yang akan direncanakan antara lain pemeliharaan korektif. Pemeliharaan ini mengkoreksi permasalahan yang ditemukan pada saat sistem digunakan. Terutama implementasi rumus metode *Euclidean Distance* dan metode *Haversine*.

5. KESIMPULAN

- Aplikasi dapat digunakan pada perangkat android
- Pengujian menunjukkan hasil sama saat perhitungan jarak antara metode *Euclidean Distance* dan metode *Haversine*

6. REFERENSI

- Nazarudin Safaat Harahap. 2012. Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung
- Rizky Muhammad Alhamad. 2015. Perancangan Rute Pengiriman BBM di Depo Balongan Dengan Menggunakan Metode Vehicle Routing Problem. theses from JIPPTUMG
- Sariyun Naja Anwar, Isworo Nugroho dan Edy Supriyanto. 2015. Model Rute dan Peta Interaktif Posyandu di Kota

Semarang menggunakan Geolocation dan Haversine Berbasis Mobile Android. Jurnal Teknologi Informasi DINAMIK Volume 20, No.1, Januari 2015 : 51-56 ISSN : 0854-9524

Svennerberg, G. 2010. Beginning Google Maps API 3. New York: Springer Science+Business Media, LLC.

Yulianto, Ramadiani, Awang Harsa Kridalaksana. Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal. Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer e-ISSN 2597-4963 dan p-ISSN 1858-4853. Vol. 13, No. 1 Februari 2018