

Pengolahan Data Spasial-Geolocation untuk Menghitung Jarak 2 Titik

Spatial-Geolocation Data Processing to Calculate 2 Point

Ahmad Hajar¹, Isnan Nabawi², Lili Kartikawati³, Fadya Rizka Yudana⁴,
Setia Budi⁵, Nanang Prasetyantara⁶

^{1,2,3,4,5,6}Magister Teknik Informatika, Universitas Amikom Yogyakarta

Email : *¹ahmad.1294@students.amikom.ac.id,

²isnan.1299@students.amikom.ac.id, ³lili.1300@students.amikom.ac.id,

⁴fadya.1318@students.amikom.ac.id, ⁵setia.1303@students.amikom.ac.id,

⁶nanang.21@students.amikom.ac.id

Abstrak

Perkembangan teknologi Informasi yang berkembang pesat sudah menyentuh ilmu pengetahuan lain seperti perkembangan Sistem Informasi Geografis (SIG). Teknologi SIG memungkinkan sistem dapat menentukan lokasinya dengan memanfaatkan banyak masukan seperti satelit, RFID, WIFI. Geolocation dapat memberikan informasi latitude dan longitude disebut juga dengan koordinat geografis yang merupakan informasi mendasar dari sebuah lokasi di bumi. Basis data dibutuhkan untuk mengembangkan SIG. MySQL sudah mendukung penyimpanan tipe data spasial yang salah satu kelasnya adalah point yang dapat menyimpan data koordinat latitude, longitude dan SRID. Penelitian bertujuan menerapkan teknologi dalam melakukan penyimpanan, memanipulasi data koordinat geografis dan pengolahannya sehingga mendapatkan informasi yang dibutuhkan. Penerapan penelitian dilakukan dengan menambahkan data lokasi beberapa rumah sakit yang ada di kota Yogyakarta, mengubah data lokasi, menampilkan rumah sakit yang terdekat dari titik tertentu, dan menampilkan rumah sakit pada radius tertentu dari titik tertentu. Penelitian membuktikan bahwa tipe data spasial yang menampung data lokasi berupa latitude dan longitude rumah sakit dapat diolah untuk memberikan informasi pencarian rumah sakit terdekat di Yogyakarta dari radius kurang dari sama dengan 3 kilo meter dari titik 110.361994, -7.764768 dalam database yang telah dimiliki.

Kata Kunci Data Spasial, Geolocation, Koordinat Geografis, MySQL.

Abstract

The development of information technology that is growing rapidly has touched other sciences such as the development of Geographical Information Systems (GIS). GIS technology allows the system to determine its location by utilizing many inputs such as satellite, RFID, WIFI. Geolocation can provide latitude and longitude information - geographic coordinates which are basic information about a location on earth. Database development to support GIS. MySQL supports the storage of spatial data types, a point class that can store latitude, longitude and SRID coordinate data. This study intends to apply geolocation technology, especially in terms of storing, manipulating geographic coordinate data and processing it so that it gets the information needed. The application of research was carried out by adding the location data of several hospitals in the city of Yogyakarta, change the location data, display the closest hospital from a certain point, and display the hospital in a certain radius from a certain point. Experiments will prove that the spatial data type can be used to obtain search information for the nearest hospital in Yogyakarta from a radius less than 3 kilometers from the point 110.361994, -7.764768 in the database that has been provided.

Keywords: Spatial Data, Geolocation, Geographical Coordinates, MySQL.

1. PENDAHULUAN

Perkembangan Teknologi Informasi yang pesat dan terus berkembang telah disinergikan dengan ilmu pengetahuan lain sehingga menghasilkan sebuah kinerja yang lebih optimal seperti perkembangan Sistem Informasi Geografis (SIG). Teknologi SIG yang sangat dibutuhkan dan banyak diimplementasikan dalam membantu pekerjaan manusia adalah teknologi *Geolocation* yang merupakan sebuah API dari HTML 5 yang diperuntukkan untuk Layanan Berbasis Lokasi (LBS). Teknologi *Geolocation* memberikan kemudahan pengguna untuk mengetahui lokasi keberadaannya berdasarkan latituded dan longitude. *Geolocation* bekerja dengan cara melacak keberadaan perangkat yang kita gunakan dan disimpulkan dengan memanfaatkan RFID, WiFi dan Bluetooth MAC alamat dan GSM/CDMA ID sel, serta masukan pengguna.

Menghitung jarak antara 2 titik di bumi dapat dilakukan dengan *Haversine Formula*. Persamaan yang dikenal akurat untuk menghitung jarak di antara 2 titik dengan memperhitungkan panjang garis lurus antara dua titik dengan mempertimbangkan kelengkungan tertentu yang dimiliki bumi. Pengguna sistem dapat mengetahui posisi keberadaannya terhadap lokasi tertentu walaupun masih mengabaikan relief bumi yang sebenarnya.

Jarak dari suatu titik koordinat / tempat merupakan data yang berkaitan dengan bidang dan ruang (spasial). Data spasial berupa titik (*point*) mampu merepresentasikan grafis yang paling sederhana pada suatu objek. Penggunaan data *spasial-point* dalam kehidupan diantaranya untuk menentukan lokasi fasilitas kesehatan, lokasi sekolah, lokasi tempat umum, dll. Pemanfaatan data spasial di kolaborasikan dengan basis data untuk mendapatkan informasi baru. Data spasial titik koordinat suatu objek ditampung dalam *fieldGeolocation* yang memuat *latitude* dan *longitude*, dimana tipe data yang dapat digunakan untuk menyimpan field tersebut memiliki 2 macam penerapan yang dapat dilakukan. Penerapan pertama untuk penyimpanan titik koordinat dari suatu objek dengan menggunakan 2 *field*, yaitu *field* latitude dan *field* longitude. Penerapan kedua dengan menggunakan tipe data *point* yang dapat menyimpan latitude dan longitude sekaligus dalam 1 *field*.

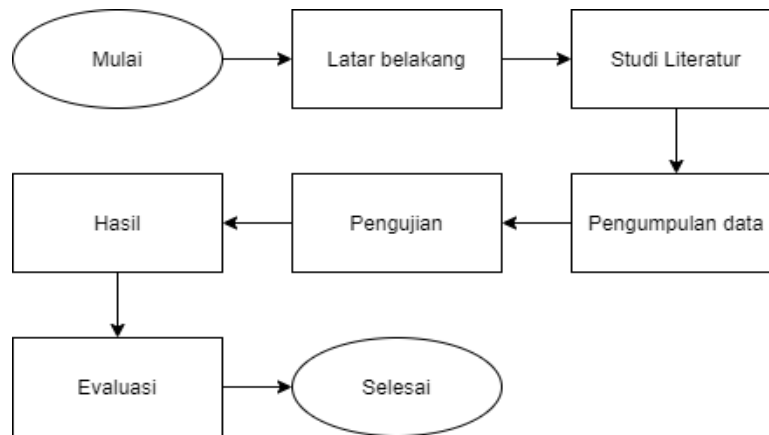
Penelitian mengenai pemanfaatan teknologi *geolocation* pada pencarian jarak lokasi pernah dilakukan dengan menerapkan formula haversine untuk pencarian lokasi terdekat [1], GoogleMaps sebagai pembangun peta digital pada sistem informasi geografis pencarian jarak terdekat, lokasi lapangan futsal di kota Samarinda, Kalimantan Timur. Menghitung jarak koordinat berdasarkan latitude dan longitude dilakukan dalam penelitian membangun aplikasi penghitung jarak koordinat berdasarkan latitude dan longitude dengan metode euclidean distance dan metode haversine dengan memanfaatkan API google maps menghasilkan kesimpulan bahwa perhitungan jarak antara metode euclidean distance dan metode haversine menunjukkan hasil yang sama [2].

Penerapan yang dilakukan pada penelitian ini adalah dengan menambahkan data lokasi beberapa rumah sakit yang ada di kota Yogyakarta, mengubah data lokasi, menampilkan rumah sakit yang terdekat dari titik tertentu, dan menampilkan rumah sakit pada radius tertentu dari titik tertentu. Penelitian ini bertujuan untuk memperlihatkan bahwa tipe data spasial dapat menyimpan data koordinat dengan baik. Percobaan akan membuktikan bahwa tipe data spasial yang menampung data lokasi berupa latitude dan longitude rumah sakit dapat diolah untuk memberikan informasi pencarian rumah sakit terdekat di Yogyakarta dari radius kurang dari sama dengan 3 kilo meter dari titik 110.361994, -7.764768 dalam database yang telah dimiliki dengan keluaran hasil pemrosesan sebanyak 3 rumah sakit.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk mengetahui kemampuan tipe data *point* untuk menyimpan data koordinat geografis. Selain itu penelitian ini juga akan mengimplementasikan data yang disimpan dengan tipe data *point* dapat memberi informasi yang berguna. Untuk melakukan implementasi tersebut Peneliti akan menggunakan beberapa data lokasi rumah sakit yang ada di

kota Yogyakarta. Peneliti menggunakan Google Maps untuk mendapatkan data lokasi tersebut. Data lokasi tersebut kemudian akan digunakan untuk mencari jarak antar rumah sakit maupun terhadap titik tertentu. Hasil implementasi yang dilakukan selanjutnya akan dievaluasi untuk mendapatkan kelebihan dan kekurangan dalam penggunaan tipe data *point* untuk menyimpan data koordinat geografis. Tahapan yang telah dijabarkan diatas sesuai dengan gambar 1 alur penelitian yang telah dirancang.



Gambar 1. Alur Penelitian

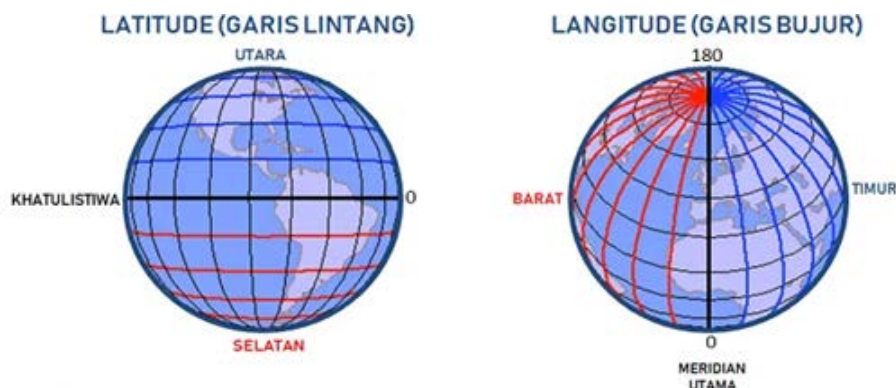
2.1. Studi Literatur

2.1.1. Data Spasial

Data spasial adalah data mengenai objek atau unsur geografis yang dapat diidentifikasi dan mempunyai acuan lokasi berdasarkan koordinat tertentu [3]. Data spasial meliputi data grafis dan data atribut. Salah satu terwujudnya SIG adalah adanya data spasial. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang berbeda dari data yang lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut). Data spasial menunjukkan posisi geografi dimana setiap karakteristik memiliki satu lokasi yang harus ditentukan dengan cara yang unik [4]. Data spasial merupakan salah satu aspek dari GIS. Data spasial memuat beberapa parameter termasuk posisi latitude dan longitude. MySQL juga memiliki tipe data Spasial untuk menyimpan parameter-parameter data spasial.

2.1.2. Geolocation

Geolocation adalah perangkat spesifik API, beberapa browser ada yang mendukungnya ada juga yang tidak, sehingga disimpulkan *Geolocation* tidak selalu bisa untuk aplikasi web [4]. Titik lokasi di bumi disederhanakan dalam posisi garis lintang yang sering disebut dengan *latitude* dan garis bujur yang sering disebut *longitude*. Garis lintang adalah garis yang menentukan jarak melintang dari kutub utara hingga kutub selatan. Garis Lintang 0° berada di khatulistiwa. Garis lintang 0° hingga 90° berada di antara garis khatulistiwa hingga kutub utara. Sedangkan 0° hingga -90° berada di antara garis khatulistiwa hingga kutub selatan bumi, seperti yang terlihat pada gambar 2 garis lintang dan bujur dibagian sisi kiri. Garis Bujur adalah garis yang menentukan jarak membujur dari barat hingga timur bumi, seperti yang terlihat pada gambar 2 garis lintang dan bujur dibagian sisi kanan. Garis bujur berada di meridian referensi atau garis bujur 0° berada di Kota Greenwich, Inggris.



Gambar 2. Garis Lintang dan Bujur

Sistem koordinat adalah posisi suatu titik [5]. Sistem Koordinat Geografis (*Geographic Coordinate System* – GCS) adalah sistem koordinat yang mengacu pada bentuk bumi sebenarnya yaitu bulat/*ellipse*. Sistem koordinat geografis ini terdiri dari garis bujur (longitude) dan garis lintang (latitude). Koordinat Geographic dibedakan menjadi tiga berdasarkan satuannya yaitu:

1. Degree, Decimal (DD, DDDD) Contoh: S 1.23456 R 102.34567
2. Degree, Minute (DD MM, MMMM) Contoh: S 1. 23,4567' E 123. 45,6789'
3. Degree, Minute, Second (DD MM SS, SS) Contoh: S 1. 23' 45,67'' E 123.45' 67,89''

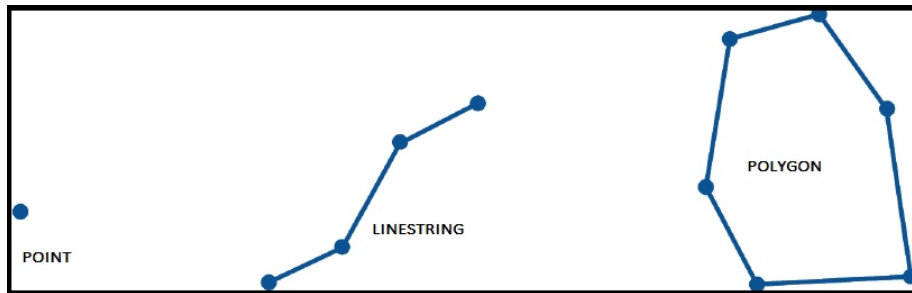
2.1.3. Menghitung Jarak antar Koordinat Geografis

Untuk menghitung jarak antar koordinat geografis berbeda dengan menentukan jarak dalam koordinat kartesian. Dalam koordinat geografis, karena bumi berbentuk hampir seperti bola maka jarak antar garis berbentuk melengkung. Cara yang tepat untuk menghitung jarak antar koordinat geografis adalah menggunakan formula Haversine. Formula Haversine adalah pernyataan matematika yang dipakai di navigasi, yang menghasilkan jarak lingkaran antara dua titik pada permukaan bola (bumi) merujuk pada bujur dan lintang [6].

2.1.4. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan datanya [7]. Saat ini perkembangan MySQL sudah mencapai versi 8. Pada MySQL versi 8 terdapat tipe data POINT yang di dalamnya dapat menyimpan informasi koordinat geografis menggunakan satuan degree decimal. POINT merupakan salah satu tipe data spasial Tipe data spasial adalah tipe data spesifik yang digunakan dalam penyimpanan data spasial [8]. Tipe data spasial dibagi menjadi 2 kelompok yaitu tipe data spasial bernilai geometri tunggal dan tipe data spasial berisi seluruh koleksi geometri. Tipe data spasial untuk geometri tunggal diantaranya adalah:

1. POINT (satu titik)
2. LINESTRING (memiliki setidaknya 2 poin)
3. POLYGON (memiliki setidaknya empat titik, poligon harus ditutup)
4. GEOMETRI (bidang data universal, dapat menyimpan POINT, LINESTRING atau POLYGON)



Gambar 3. POINT, LINESTRING dan POLYGON

Point merupakan salah satu class dari tipe data spasial di MySQL. Kelas-kelas dari tipe data spasial sendiri terdiri dari Geometry, Point, Linestring dan Polygon. Pembahasan disini berfokus pada objek point. Point adalah geometri yang mewakili satu lokasi dalam ruang koordinat. Bayangkan peta dunia berskala besar dengan banyak kota. Objek Point dapat mewakili setiap kota. Di peta kota, objek Point bisa mewakili halte bus. Point memiliki 3 properti, yaitu:

1. Nilai koordinat X.
2. Nilai koordinat Y.
3. Titik didefinisikan sebagai geometri dimensi-nol (SRID=0).

2.2. Pengumpulan Data

Data yang dibutuhkan adalah data koordinat geometris rumah sakit yang ada di kota Yogyakarta. Tabel 1 koordinat geografis rumah sakit di kota yogyakarta adalah pencatatan data koordinat geometris rumah sakit yang ada di kota yogyakarta. Koordinat geometris tersebut diambil menggunakan aplikasi Google Maps.

Tabel 1. Koordinat Geografis Rumah Sakit di Kota Yogyakarta

No	Nama Rumah Sakit	Longitude	Latitude
1	Sardjito	110.3712855	-7.7686163
2	JIH	110.4033776	-7.7576375
3	Panti Rapih	110.3775519	-7.7770367
4	Siloam	110.3908771	-7.7834413
5	Bethesda	110.3776914,	-7.7839887
6	RSUD Yogyakarta	110.3780079	-7.8258262
7	PKU Muhammadiyah	110.3623116	-7.8007412

3. HASIL DAN PEMBAHASAN

Pengumpulan data koordinat geografis rumah sakit di kota yogyakarta telah dilakukan dengan pendataan seperti yang terlihat pada tabel 1 koordinat geografis rumah sakit di kota yogyakarta. Tahapan selanjutnya adalah melakukan pengujian kemampuan tipe data point untuk dapat menyimpan titik koordinat suatu lokasi dan apa keluaran informasi yang didapat dari pengolahan data spasial titik koordinat yang disimpan dalam tipe data point tersebut. Tahapan yang akan diberikan disini adalah sebagai berikut:

3.1. Membuat Tabel Lokasi

Koordinat geografis rumah sakit di kota yogyakarta yang telah dikumpulkan selanjutnya akan dicatat dalam sebuah tabel penyimpanan database manajemen sistem (DBMS) agar mudah dalam pengaksesan dan pengelolaan data. Pada pengujian ini digunakan DBMS MYSQL versi 8 yang telah dilengkapi tipe data point. Pembuatan tabel rumah_sakit untuk dipakai sebagai tempat

penyimpanan data spasial koordinat geografis rumah sakit, berikut adalah perintah query untuk membuat tabel rumah_sakit:

```
CREATETABLE`rumah_sakit` (
  `id`intunsignedNOTNULLAUTO_INCREMENT,
  `name`varchar(100) NOTNULL,
  `location`pointNOTNULL,
  PRIMARYKEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Query pembuatan tabel rumah_sakit diatas terlihat bahwa field name bertipe data varchar dengan panjang yang disediakan 100, pada field location memakai tipe data point untuk menyimpan koordinat geografis rumah sakit yaitu longitude dan latitude dan field id sebagai primary key.

3.2. Insert Data Lokasi

Tabel rumah_sakit yang telah dibuat ditahapan diatas selanjutnya akan dilakukan pengujian. Tipe data yang menjadi pilihan untuk dapat menyimpan koordinat geografis rumah sakit adalah point. Pada kesempatan ini akan dilakukan pencatatan kedalam tabel rumah_sakit dari data koordinat geografis yang telah dikumpulkan. Pencatatan kedalam tabel rumah_sakit dari data koordinat geografis rumah sakit yang ada di kota yogyakarta seperti yang terlihat pada tabel 1 koordinat geografis rumah sakit di kota yogyakarta dapat dilihat pada query insert data lokasi dibawah ini:

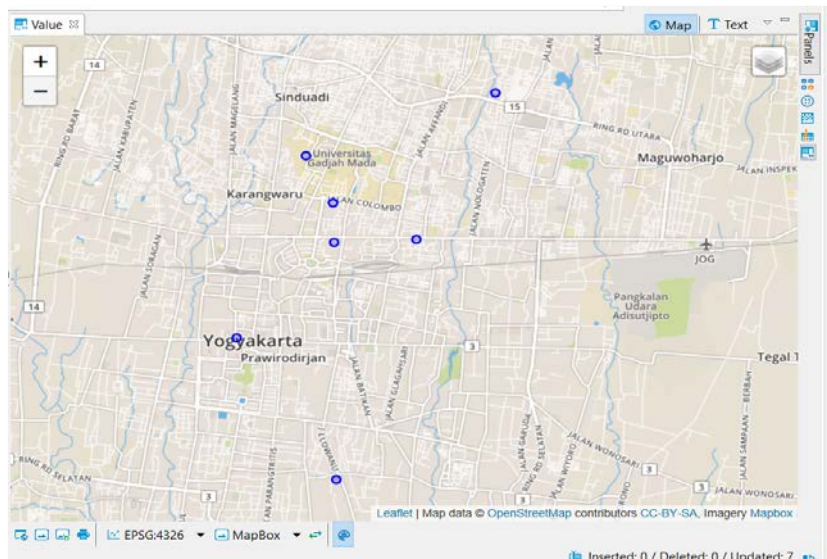
```
INSERTINTO`rumah_sakit` (`name`, `location`)
VALUES ('Sardjito', POINT(110.3712855, -7.7686163)),
 ('JIH', POINT(110.4033776, -7.7576375)),
 ('Panti Rapih', POINT(110.3775519, -7.7770367)),
 ('Siloam', POINT(110.3908771, -7.7834413)),
 ('Bathesda', POINT(110.3776914, -7.7839887)),
 ('RSUD Yogyakarta', POINT(110.3780079, -7.8258262)),
 ('PKU Muhammadiyah', POINT(110.3623116, -7.8007412));
```

Query penyisipan data di tabel rumah sakit diatas, memasukan data pada field name berupa nama-nama rumah sakit seperti Sardjito, JIH, Panti Rapih, Siloam, Bethesda, RSUD Yogyakarta, PKU Muhammadiyah dan memasukan data location dari rumah sakit yang namanya telah disebutkan sebelumnya berupa longitude dan latitude.

Keberhasilan tabel rumah_sakit dalam menyimpan data koordinat geografis rumah sakit, terutama pada field location yang menggunakan tipe data point, dimana tipe data tersebut dapat menampung longitude dan latitude lokasi rumah sakit sehingga hasilnya dapat terlihat pada gambar 4 hasil query data lokasi dan penampakan pada aplikasi Map dari rumah sakit yang telah dihasilkan dari query dapat dilihat pada gambar 5 merupakan Map hasil query data lokasi dibawah ini:

	id	name	location
1	1	Sardjito	POINT (110.3732855 -7.7686163)
2	2	JIH	POINT (110.4033776 -7.7576375)
3	3	Panti Rapih	POINT (110.3775519 -7.7770367)
4	4	Siloam	POINT (110.3908771 -7.7834413)
5	5	Bathesda	POINT (110.3776914 -7.7839887)
6	6	RSUD Yogyakarta	POINT (110.3780079 -7.8258262)
7	7	PKU Muhammadiyah	POINT (110.3623116 -7.8007412)

Gambar 4. Hasil Query Data Lokasi



Gambar 5. Map Hasil Query Data Lokasi

3.3. Update Data Lokasi

Pengelolaan data tidak terlepas dari pembaharuan data, pada kesempatan ini akan ditunjukkan bahwa tipe data point yang telah dipilih sebagai tempat penyimpanan koordinat geografis rumah sakit, mendukung untuk dilakukan pembaharuan data. Update data koordinat geografis rumah sakit menggunakan perintah query update data lokasi seperti di bawah ini:

```
UPDATE hospitals.rumah_sakit
SET location = POINT(110.3732855, -7.7686163)
WHERE id = 1;
```

Query diatas akan bekerja mencari database hospital pada tabel rumah_sakit dengan id = 1 untuk dilakukan pembaharuan pada isian field locationnya.

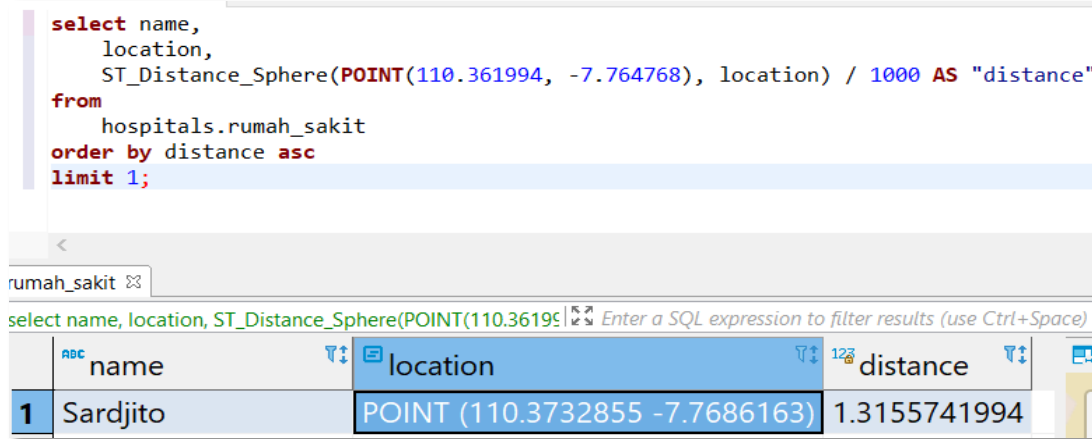
3.4. Menampilkan Rumah Sakit Terdekat dari Titik Tertentu

Pengolahan data spasial koordinat geografis yang menggunakan tipe data point ini akan mampu diolah untuk menghasilkan keluaran informasi terkait geografi – mencari rumah sakit terdekat dari suatu koordinat geografis lokasi tertentu. Untuk menampilkan rumah sakit terdekat dari titik tertentu, asumsi yang menjadi lokasi acuan pencarian adalah berada dititik 110.361994, -7.764768, maka perintah query pencariannya dapat dilihat dibawah ini:

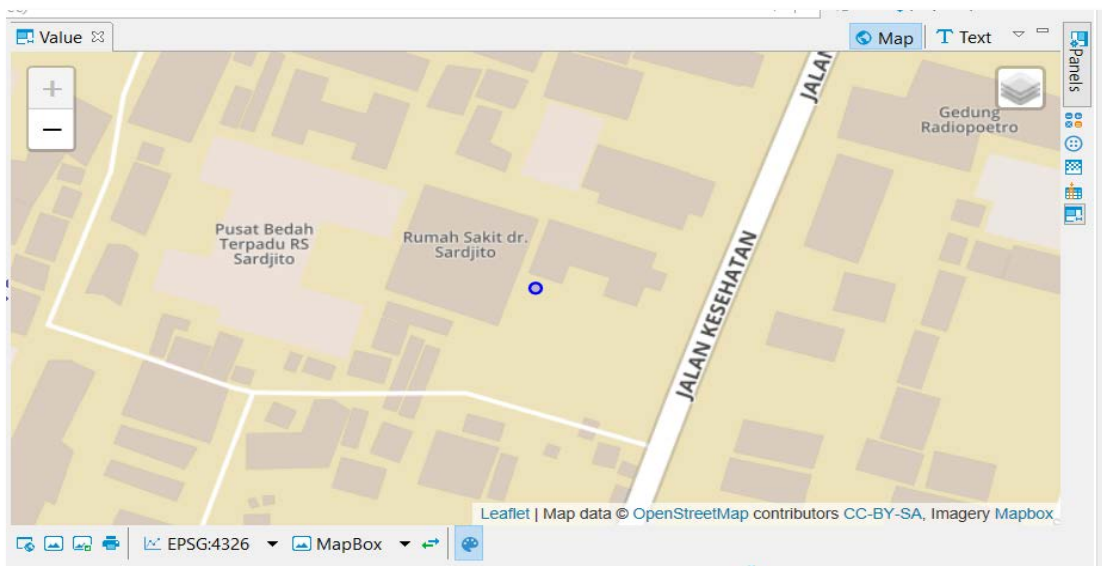
```
select name,
location,
ST_Distance_Sphere(POINT(110.361994, -7.764768), location) /
1000AS"distance"
fromhospitals.rumah_sakit
orderby distance asc
limit1;
```

Query diatas akan menyeleksi data pada field name dan location di database hospital pada tabel rumah_sakit untuk menghitung jarak rumah sakit terdekat dari titik yang di set berada di longitude 110.361994 dan latitude -7.764768 dalam kilo meter (/1000) ditampung sebagai “distance” kemudian diurutkan berdasarkan “distance” dari jarak kecil ke besar dan dibatasi 1 untuk keluarannya.

Informasi keluaran dari perintah query pencarian rumah sakit terdekat dari titik tertentu dapat dilihat pada gambar 6 merupakan hasil query data rumah sakit terdekat dari titik tertentu dan penampakan pada aplikasi Map dari hasil pencarian rumah sakit terdekat dari titik tersebut dapat dilihat di gambar 7.



Gambar 6. Hasil Query Data Rumah Sakit Terdekat dari Titik Tertentu



Gambar 7. Map Hasil Query Data Rumah Sakit Terdekat dari Titik Tertentu

3.5. Menampilkan Rumah Sakit yang Lokasinya Berada di Radius Tertentu

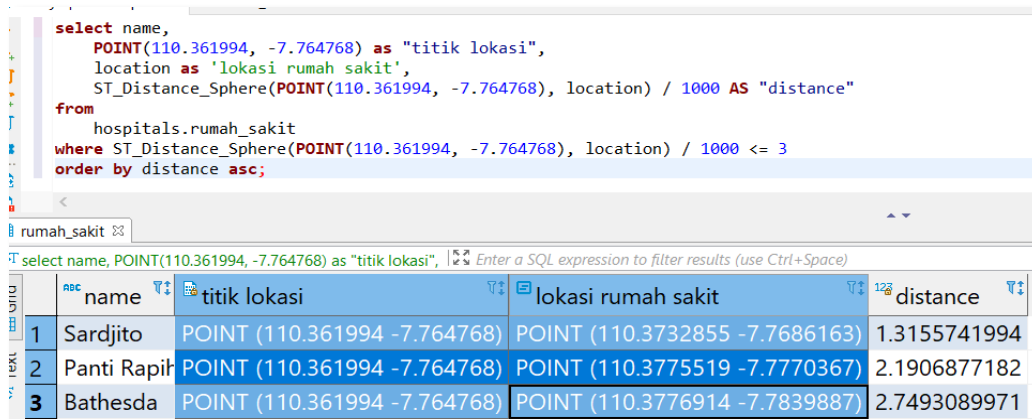
Pengolahan data spasial koordinat geografis yang dilakukan pada tahap sebelumnya telah dihasilkan keluaran informasi rumah sakit terdekat dari suatu titik tertentu, pada tahapan selanjutnya akan dilakukan pengujian pengolahan data spasial bertipe data point untuk dapat menghasilkan keluaran informasi rumah sakit di radius tertentu dari titik tertentu. Diberikan suatu persepsi untuk menampilkan rumah sakit di radius tertentu misal kurang dari sama dengan 3 kilometer dari sebuah titik lokasi, perintah query untuk mencari rumah sakit yang berada di radius kurang dari sama dengan 3 kilo meter dari titik 110.361994, -7.764768 adalah sebagai berikut:


```

select name,
POINT(110.361994, -7.764768) as "titik lokasi",
location as 'lokasi rumah sakit',
ST_Distance_Sphere(POINT(110.361994, -7.764768), location) /
1000 AS "distance"
from
hospitals.rumah_sakit
where ST_Distance_Sphere(POINT(110.361994, -7.764768), location) / 1000 <=
3
order by distance asc;

```

Informasi keluaran dari perintah query pencarian rumah sakit dari radius kurang dari sama dengan 3 kilo meter dari titik 110.361994, -7.764768 dapat terlihat pada gambar 8. Tampilan Map dari hasil pencarian rumah sakit dari radius tersebut dari titik yang telah ditentukan dapat dilihat pada gambar 9.



The screenshot shows a SQL query editor with the following query:

```

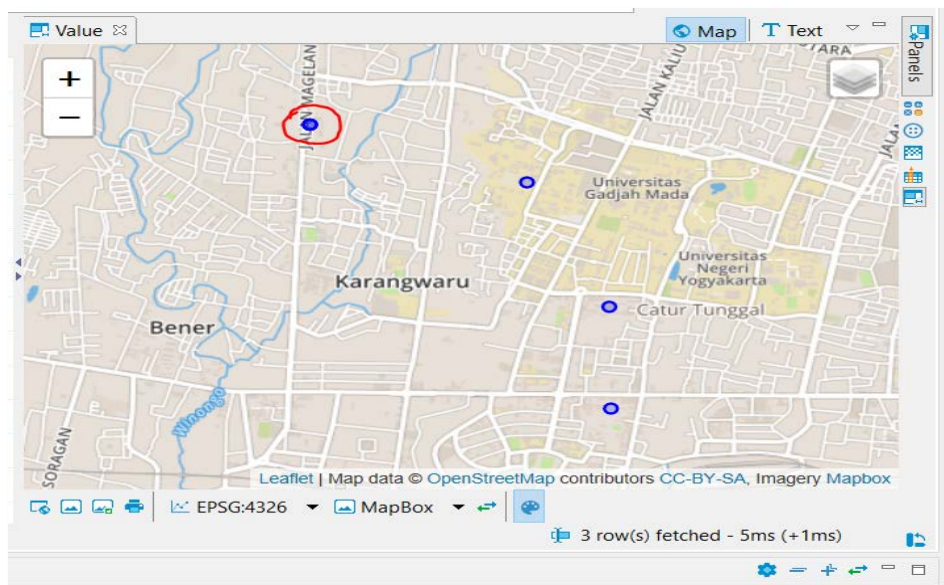
select name,
POINT(110.361994, -7.764768) as "titik lokasi",
location as 'lokasi rumah sakit',
ST_Distance_Sphere(POINT(110.361994, -7.764768), location) / 1000 AS "distance"
from
hospitals.rumah_sakit
where ST_Distance_Sphere(POINT(110.361994, -7.764768), location) / 1000 <= 3
order by distance asc;

```

Below the query, a table displays the results of the query:

	name	titik lokasi	lokasi rumah sakit	distance
1	Sardjito	POINT (110.361994 -7.764768)	POINT (110.3732855 -7.7686163)	1.3155741994
2	Panti Rapih	POINT (110.361994 -7.764768)	POINT (110.3775519 -7.7770367)	2.1906877182
3	Bathesda	POINT (110.361994 -7.764768)	POINT (110.3776914 -7.7839887)	2.7493089971

Gambar 8. Hasil Query Data Lokasi dengan Radius Tertentu



Gambar 9. Map Hasil Query Data Lokasi dengan Radius Tertentu

Gambar 9 map hasil *query* data lokasi memperlihatkan lingkaran merah yang menandai lokasi yang menjadi pusat titik perhitungan radius, radius yang dijadikan persyaratan adalah 3 kilo meter sehingga terlihat jelas perbandingan jarak titik lokasi terhadap rumah sakit yang terdapat diradius tersebut. Titik biru lainnya adalah lokasi rumah sakit di radius kurang dari sama dengan 3 kilometer.

4. KESIMPULAN

Kemampuan penyimpanan data koordinat geografis dari rumah sakit di kota Yogyakarta dengan pilihan tipe data *point* terbukti telah dapat menyimpan longitude dan latitude dari koordinat geografis rumah sakit di Kota Yogyakarta.

Pemanfaatan Sistem Informasi Geografis (SIG) dalam menyelesaikan permasalahan nyata keruangan seperti jarak suatu lokasi dengan lokasi yang lain, keteraksesan fasilitas umum dari suatu titik lokasi dan lain-lainnya. Pengelolaan data spasial koordinat geografis *per-type point* yang telah dilakukan pada penelitian ini, mampu memberikan keluaran informasi rumah sakit terdekat dari suatu titik tertentu dan informasi rumah sakit dari radius tertentu dari suatu titik tertentu. Berdasarkan percobaan yang sudah dilakukan, tipe data spasial dapat menyimpan data koordinat dengan baik. Percobaan membuktikan bahwa tipe data spasial dapat digunakan untuk mendapatkan informasi pencarian rumah sakit terdekat dari radius kurang dari sama dengan 3 kilo meter dari titik 110.361994, -7.764768 yaitu sebanyak 3 rumah sakit.

5. SARAN

Pemenuhan kebutuhan informasi baru terkait data keruangan semakin besar, oleh karena itu dibutuhkan kemampuan pengolahan data yang cepat dan akurat. Pemrosesan informasi baru yang melibatkan kegiatan perhitungan dan pengambilan data yang lebih cepat diperlukan pendekatan tipe data sesuai kebutuhan. Kinerja yang dilakukan oleh tipe data berindeks akan memberikan efek yang jauh lebih besar dari pada menjaga ruang penyimpanan minimal. Perlu dilakukan pengukuran kecepatan terhadap proses pengolahan informasi baru yang melibatkan tipe data yang digunakan. Kinerja dari tipe data tersebut yang tidak hanya memiliki penyimpanan yang minimal tetapi harus mampu memberikan kinerja yang maksimal. Perlu dilakukan pengujian untuk mengukur kinerja dari suatu tipe data spasial agar pengolahan data yang cepat dapat dihasilkan.

DAFTAR PUSTAKA

- [1] Yulianto., Ramadiani., Kridalaksana, A. H., 2018, Penerapan Formula Haversine pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal, *Jurnal Ilmiah Ilmu Komputer*, No.1, Vol. 13, Hal. 14-15.
- [2] Pamungkas, C. A., 2019, Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude dan Longitude dengan Metode Euclidean Distance dan Metode Haversine, *Jurnal Informa Politeknik Indonusa Surakarta*, No.2, Vol. 5, Hal. 8-13.
- [3] Zendrato N., Ginting, S. B. F., 2017, Sistem Informasi Geografis Pemetaan Rumah Makan Studi Kasus pada Wilayah Medan Kabanjane, *Jurnal Ilmiah Teknologi Informasi Neumann*, No.2, Vol. 2, Hal. 24-29.
- [4] Handayani, D., Soelistijadi, R., Sunardi., 2005, Pemanfaatan Analisis Spasial untuk Pengolahan Data Spasial Sistem Informasi Geografi, *Jurnal Teknologi Informasi DINAMIK*, No. 2, vol. 10, Hal. 108-116.

- [5] Prahasta, E., 2009, Sistem Informasi Geografis: Konsep-konsep Dasar (Perspektif Geodesi & Geomatika), *Penerbit Informatika*, Bandung
 - [6] Setiawan, H., 2014, Implementasi Haversine Formula Pada Lokasi, *Jurnal Fahma*
 - [7] Arief, M. R., 2011, Pemrograman Web Dinamis Menggunakan PHP & MySQL, *Penerbit Andi Offset*, Yogyakarta
 - [8] Steinhoff, S., 2019, Procedural Methods for the Implementation of Species Range Maps in the Web-based Zoological Database snakeDB.org, *Thesis*, Master of Science in Information Technology, University of southern Denmark, Kolding.
-