

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal local yaitu:

1. Penelitian yang dilakukan oleh Fahrul Agus, Dkk (2017) dengan judul Sistem Informasi Geografi untuk Simulasi Rute Penerbangan Virtual. Dimana dalam penelitian yang di lakukan oleh penulis menjelaskan bahwa Studi ini telah menerapkan teknik BDP dan aturan penerbangan VFR pada sistem simulasi penerbangan virtual. Metode pemrograman dinamis mundur yang diterapkan dalam sistem dapat menyelesaikan masalah pencarian rute penerbangan VFR dengan nilai kesuksesan akhir mencapai 86.19% dari seluruh 567 rute yang melalui bandara penghubung. Penelitian ini memiliki perbedaan dengan penelitian yang di angkat oleh penulis bahwa penulis menerapkan metode *Spherical Law Of Cosine* yang digunakan sebagai pencarian lokasi dari sebuah *node* awal ke *node* tujuan yang ingin masyarakat kunjungi, sehingga dengan adanya sistem ini dapat memberikan manfaat pada masyarakat dalam pencarian lokasi.
2. Penelitian yang dilakukan oleh Koko Mukti Wibowo, Dkk (2015) dengan judul SISTEM INFORMASI GEOGRAFIS (SIG) MENENTUKAN LOKASI PERTAMBANGAN BATU BARA DI PROVINSI BENGKULU BERBASIS

ANDROID. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa bahasa pemrograman *ANDROID* dapat memberikan kemudahan dalam perancangan Sistem Informasi Geografis Pusat Pertambangan di Provinsi Bengkulu. Dan database *MySQL* dapat menampung informasi dan data yang ada Dinas Energi dan Mineral Provinsi Bengkulu. Dengan adanya sistem ini dapat memudahkan pengguna dalam pencarian lokasi tambang di Provinsi Bengkulu. Penelitian ini memiliki beberapa perbedaan dengan penelitian yang di angkat oleh penulis bahwa dengan adanya sistem ini dapat memberikan manfaat pada masyarakat dalam pencarian lokasi.

3. Penelitian yang dilakukan oleh Cecep J Abbas, Dkk (2014) dengan judul SISTEM PEMETAAN LAHAN PERTANIAN DI KABUPATEN KUNINGAN. Dimana dalam penelitian yang di lakukan oleh penulis menjelaskan bahwa Sistem ini dapat mendeteksi lahan-lahan pertanian di Kabupaten Kuningan. Sistem ini dapat dijadikan alternatif acuan bagi setiap orang yang ingin mengetahui macam-macam komoditas pertanian yang tersebar di Kabupaten Kuningan yang akhirnya bisa untuk pengembangan komoditas itu sendiri. Berdasarkan uraian diatas, penelitian ini memiliki perbedaan dengan penelitian yang di angkat oleh penulis bahwa penulis menerapkan metode *Spherical Law Of Cosine* yang digunakan sebagai pencarian lokasi dari sebuah *node* awal ke *node* tujuan yang ingin masyarakat kunjungi, sehingga dengan adanya sistem ini dapat memberikan manfaat pada masyarakat dalam pencarian lokasi.

4. Penelitian yang dilakukan oleh Satriaji Cahyo Nugroho, dkk, 2015 dengan judul Aplikasi Pencarian Rute Perguruan Tinggi Berbasis *Android* Menggunakan *Location Based Service* (LBS) di Kota Semarang. Berdasarkan penelitian yang dilakukan, didapat hasil bahwa aplikasi pencarian rute dapat memudahkan pengguna dalam mendapatkan informasi dan menentukan rute karena data informasi dan letak lokasi-lokasi perguruan tinggi telah didaftarkan oleh peneliti. Penelitian ini memiliki perbedaan dengan penelitian yang di angkat oleh penulis bahwa penulis memanfaatkan *Google Maps* berbeda dengan penelitian yang di lakukan oleh satriaji yang menggunakan *Location Based Service* (LBS).
5. Penelitian yang dilakukan oleh (Mukhlis, dkk, 2017) dengan judul Aplikasi *Android* Peta Sekolah Bengkalis Menggunakan *Google Map API*. Berdasarkan penelitian yang di lakukan, penulis membuat sebuah sistem yang dapat menampilkan letak atau lokasi sekolah dan menampilkan informasi sekolah yang ada di Bengkalis. dapat diakses secara umum tanpa ada batasan waktu dan tempat. Aplikasi peta sekolah dapat dijalankan di berbagai *type* perangkat *mobile*, dan tampilan peta serta menampilkan informasi sesuai yang diinginkan. Adapun beberapa perbedaan penelitian yang di angkat dengan penelitian sebelumnya, selain perbedaan dengan objek penelitian metode yang digunakan juga berbeda dengan metode yang digunakan oleh penelitian.

II.2. Uraian Teoritis

Penelitian yang dilakukan dengan cara langsung pada pencarian lokasi distributor pupuk adalah menggunakan beberapa aplikasi atau bahasa program, yaitu :

II.2.1. Sistem Informasi Geografis

Sistem Informasi Geografis (GIS) adalah suatu sistem informasi menyangkut keberadaan obyek di permukaan bumi berikut informasi yang terkandung di dalamnya yang mempunyai keterkaitan secara geografis dengan obyek lainnya. Dengan sistem ini data dapat dikelola, dilakukan manipulasi untuk keperluan analisis secara komprehensif. [3].

SIG adalah sistem komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi-informasi yang berhubungan dengan permukaan bumi. Pada dasarnya, istilah sistem informasi geografi merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografi. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG.[4].

Sistem Informasi Geografi (SIG) adalah sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi geografis. Sistem ini dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang

memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis yaitu [5] :

- a) Masukan,
- b) Manajemen data (penyimpanan dan pemanggilan data),
- c) Analisis dan manipulasi data, dan
- d) Keluaran

Ciri-ciri Sistem Informasi Geografis adalah sebagai berikut:

- a) SIG memiliki sub sistem input data yang menampung dan dapat mengolah data spasial dari berbagai sumber. Sub sistem ini juga berisi proses transformasi data spasial yang berbeda jenisnya, misalnya dari peta kontur menjadi titik ketinggian.
- b) SIG mempunyai subsistem penyimpanan dan pemanggilan data yang memungkinkan data spasial untuk dipanggil, diedit, dan diperbaharui.
- c) SIG memiliki subsistem manipulasi dan analisis data yang menyajikan peran data, pengelompokan dan pemisahan, estimasi parameter dan hambatan, serta fungsi permodelan.
- d) SIG mempunyai subsistem pelaporan yang menyajikan seluruh atau sebagian dari basis data dalam bentuk tabel, grafis dan peta.[4].

Subsistem Sistem Informasi Geografis Subsistem yang dimiliki oleh SIG yaitu data input, data output, data management, data manipulasi dan analisis. Subsistem SIG tersebut dijelaskan dibawah ini:

- a) Data Input: Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan data atribut dari berbagai sumber. Subsistem ini pula yang bertanggung jawab dalam mengkonversi atau mentransformasi format data-data aslinya ke dalam format yang digunakan oleh SIG.
- b) Data Output: Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk softcopy maupun bentuk hardcopy seperti: tabel, grafik, peta dan lain-lain.
- c) Data Management: Subsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, dan diedit.
- d) Data manipulasi dan analisis: Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan permodelan data untuk menghasilkan informasi yang diharapkan.[4]

SIG merupakan sistem kompleks yang biasanya terintegrasi dengan lingkungan sistem-sistem komputer yang lain di tingkat fungsional dan jaringan. Menurut Gistut, komponen SIG terdiri dari perangkat keras, perangkat lunak, data dan informasi geografi, serta manajemen. Komponen SIG dijelaskan di bawah ini :

- a) Perangkat keras (Hardware): Pada saat ini SIG tersedia untuk berbagai platform perangkat keras mulai dari PC desktop, workstations, hingga multiuser host yang dapat digunakan oleh banyak orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan (harddisk) yang besar, dan mempunyai kapasitas memori

(RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat terhadap karakteristik-karakteristik fisik perangkat keras ini sehingga keterbatasan memori pada PC30 pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk SIG adalah komputer (PC), mouse, digitizer, printer, plotter, dan scanner.

- b) Perangkat lunak (Software): Bila dipandang dari sisi lain, SIG juga merupakan sistem perangkat lunak yang tersusun secara modular dimana basisdata memegang peranan kunci. Setiap subsistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul, hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program yang masing-masing dapat dieksekusi sendiri.
- c) Data dan Informasi Geografi: SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan cara mengimport-nya dari perangkat lunak SIG yang lain maupun secara langsung dengan cara mendigitasi data spasialnya dari peta dan memasukkan data atributnya dari table-tabel dan laporan dengan menggunakan keyboard.
- d) Manajemen: Suatu proyek SIG akan berhasil jika dimanage dengan baik dan dikerjakan oleh orang-orang memiliki keahlian yang tepat pada semua tingkatan.[4].

II.2.2. JQuery

JQuery Mobile adalah framework yang berbasis *HTML/CSS* dan *javascript* untuk aplikasi *web* yang dijalankan di perangkat *mobile*. Dalam membangun web

biasa menggunakan *framework jquery*, begitu juga apabila ingin membuat aplikasi web untuk perangkat mobile (*smartphone/tablet pc*) bisa digunakan *jquery mobile*. Penggunaan *jquery mobile* akan menghasilkan tampilan yang menyesuaikan dengan lebar layar. *Script jquery mobile* merupakan bawaan dari framework *jquery mobile*, seperti *jqury.mobile-1.orc1.min.css* dan *styles.css*. Untuk menggunakan *jQuery Mobile*, pertama perlu memasukkan tiga *file* yaitu [2].

1. *JQuery CSS Mobile file (jquery.mobile-1.0a1.min.css)*
2. *Perpustakaan jQuery (jquery1.4.3.min.js)*
3. *Mobile perpustakaan jQuery (jquery.mobile-1.0a1.min.js)*.

II.2.3. Android

Android adalah sistem operasi berbasis *Linux* yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, *smartphone* dan juga *PC tablet*. Secara umum *Android* adalah platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak piranti bergerak [2].

Android dipuji sebagai platform mobile pertama yang lengkap, terbuka, dan bebas. Lengkap dalam artian android menyediakan tools dalam membangun Software dan memungkinkan peluang untuk membangun aplikasi, terbuka bagi pengembang untuk mengembangkan aplikasi android, dan bebas biaya keanggotaan, bebas biaya pengujian, tidak ada kontrak yang diperlukan, dan

aplikasi android dapat didistribusikan dan diperdagangkan dalam bentuk apapun [2].

Android adalah sistem operasi untuk handphone yang berbasis linux dengan menggunakan bahasa pemrograman java, dimana didalamnya terdapat sistem operasi, middleware, dan key application.[2]

II.2.4. Penerapan Metode *Spherical Law Of Cosine*

Metode *spherical law of cosine* diciptakan ketika tingkat presisi hasil penghitungan masih sangat terbatas. Namun sekarang, penghitungan komputer dapat memberikan tingkat presisi yang sangat akurat sehingga dengan menggunakan rumus *spherical law of cosine* sederhana, kita dapat menentukan posisi dengan cukup akurat.[5].

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

Keterangan :

R = jari-jari bumi sebesar 6371(km)

d = jarak (km).

Contoh kasus :

Perhitungan jarak koordinat Distributor Pupuk Organik Nusa Medan

Perhitungan jarak koordinat antara titik koordinat (N. 3.637603°, E

98.666044°) ke koordinat N 3.585672°, E 98.673149 °):

Radians = 0.0174533

Latitude1 = 3.637603

$$\text{Latitude2} = 3.585672$$

$$\text{Longitude1} = 98.666044$$

$$\text{Longitude2} = 98.673149$$

$$\text{Radians Latitude1} = 0.0174533 * 3.637603 = 0.0634881764$$

$$\text{RadiansLatitude2} = 0.0174533 * 3.585672 = 0.0625818091$$

$$\text{Radians Longitude1} =, 0.0174533 * 98.666044= 1.7220480657$$

$$\text{Radians Longitude2} = 0.0174533 * 98.673149 = 1.7221720714$$

$$R = 6371$$

$$d = a \text{COS} (\text{SIN}(\theta_1) * \text{SIN}(\theta_2) + \text{COS}(\theta_1) * \text{COS}(\theta_2) * \text{COS}(\lambda_2 - \lambda_1)).R$$

$$d = a \text{COS} (\text{SIN}(0.0634881764) * \text{SIN}(0.0625818091) + \text{COS}$$

$$(0.0634881764) * \text{COS}(0.0625818091) * \text{COS}(1.7221720714 - 1.7220480657$$

$$) * 6371 = 5.8280482891272$$

II.2.5. *Global Positioning System (GPS)*

Google Maps adalah layanan pemetaan berbasis *Android* service yang disediakan oleh Google dan bersifat gratis, yang memiliki kemampuan terhadap banyak layanan pemetaan berbasis *Android*. *Google Maps* juga memiliki sifat server side, yaitu peta yang tersimpan pada server Google dapat dimanfaatkan oleh pengguna. *Google Maps* API adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di *Google Maps* sesuai kebutuhan. Untuk membangun aplikasi yang memanfaatkan *Google Maps* di desktop dan mobile device maka akan digunakan *Google Maps* Javascript API v3 yang memiliki keunggulan lebih cepat dari versi sebelumnya.[3].

II.2.6. Pengertian *MySQL*

MySQL merupakan suatu jenis *database server* yang sangat terkenal. *MySQL* termasuk jenis RDBMS (*Relational Database Manajement System*). *MySQL* mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya *SQL* memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. *MySQL* merupakan RDBMS (*Relational Database Management System*) *server*. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya.[4].

II.2.7. Pengertian *Eclipse*

Eclipse adalah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat di jalankan pada semua *Platform*. *Eclipse* adalah *software Development environment* multi bahasa yang berfitur *extensible plugin* karena mengembangkan beberapa jenis aplikasi misalnya Java, Ada, C, C++ atau *Python*. [4].

II.2.8. UML (*Unified Modelling Language*)

Menurut (Windu Gatta, 2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


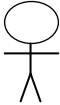
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.


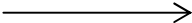
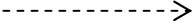
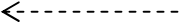
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu:

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa





	peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

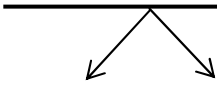
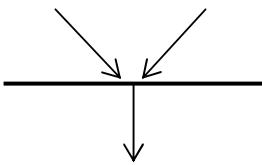
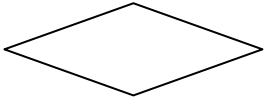

(Sumber : Windu Gatta : 4-6)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan

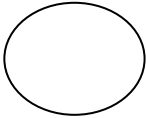
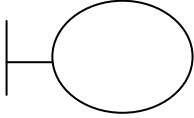
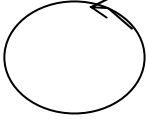

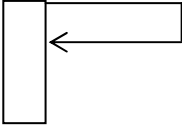


	kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gatta : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gatta : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gatta : 8-9)