

## **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 Fitri Maharany, Dkk (2015) dengan judul PENGEMBANGAN SISTEM PENCARIAN LOKASI KULINER DI KOTA MALANG DENGAN LOCATION BASED SERVICE. Dimana dalam penelitian yang dilakukan oleh penulis menjelaskan bahwa Sistem pencarian lokasi kuliner yang dibangun memberikan fasilitas kepada user untuk mengetahui lokasi kuliner yang ada di sekitar posisi user, dengan radius yang ditentukan, mencari lokasi kuliner berdasarkan kategori dan rating, memperoleh navigasi rute ke lokasi kuliner yang diinginkan, memberi rating ke lokasi kuliner dan memberi ulasan atau review lokasi kuliner.
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 WEBSITE. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa bahasa pemrograman PHP 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.

3. Penelitian yang dilakukan oleh Cecep J Abbas, Dkk (2014) dengan judul SISTEM PEMETAAN LAHAN PERTANIAN DI KABUPATEN KUNINGAN. Dimana dalam penelitian yang dilakukan 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.
4. Penelitian yang dilakukan oleh Fahrul Agus, Dkk (2016) dengan judul SISTEM INFORMASI GEOGRAFI UNTUK SIMULASI RUTE PENERBANGAN VIRTUAL. Berdasarkan penelitian yang dilakukan oleh penulis menjelaskan bahwa 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.
5. Penelitian yang dilakukan oleh Nuraniah Muslim, Dkk (2012) dengan judul SISTEM INFORMASI GEOGRAFIS BERBASIS WEB PEMETAAN POTENSI PANAS BUMI DI INDONESIA MENGGUNAKAN GOOGLE MAPS. Aplikasi Sistem Geografis ini dapat menunjukkan lokasi-lokasi yang memiliki potensi panas bumi di Indonesia, dari lokasi yang masih dalam tahap penelitian, sampai lokasi yang telah dieksplorasi.

## **II.2. Uraian Teoritis**

Penelitian yang dilakukan dengan cara langsung pada pencarian lokasi penjualan Batu Alam adalah menggunakan beberapa teoritis dan bahasa program, yaitu :

### **II.2.1. Sistem Informasi Geografis**

Menurut (Fitri Maharany, Dkk : 2015) Sistem Informasi Geografi (SIG) 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.

Menurut (Prahasta : 2002:55) 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.

### **II.2.2. Data Spasial**

Menurut (Cecep J Abbas, Dkk : 2014) Data Spasial adalah data hasil pengukuran, pencatatan, dan pencitraan terhadap suatu unsur keruangan yang berada di bawah, pada, atau di atas permukaan bumi dengan posisi keberadaannya mengacu pada system koordinat nasional.

### II.2.3. Pengertian *PHP*

Menurut tim EMS (2012:61) *PHP* adalah bahasa pelengkap HTML yang memungkinkan dibuatnya aplikasi dinamis yang memungkinkan adanya pengolahan data dan pemrosesan data. Semua sintax yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. Kemudian merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirimkan ke client, tempat pemakai menggunakan browser.

*PHP* dikenal sebagai sebuah bahasa scripting, yang menyatu dengan tag-tag HTML, dieksekusi di *server*, dan digunakan untuk membuat halaman web yang dinamis seperti halnya *Active Server Pages* (ASP) atau *Java Server Pages* (JSP). *PHP* merupakan sebuah *software Open Source*. .

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

Menurut (Cecep J Abbas, Dkk : 2014) 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.

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

Keterangan :

R = jari-jari bumi sebesar 6371(km)

$d = \text{jarak (km)}$ .

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

Menurut (Fitri Maharany, Dkk : 2015) *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.

### II.2.6. *Pengertian MySQL*

Menurut Kurniawan (2016:16) *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.

### **II.2.7. Pengertian Peta**

Menurut (Cecep J Abbas, Dkk : 2014) Peta adalah gambaran sebagian atau seluruh muka bumi baik yang terletak di atas maupun di bawah permukaan dan disajikan pada bidang datar pada skala dan proyeksi tertentu (secara matematis). Karena dibatasi oleh skala dan proyeksi maka peta tidak akan pernah selengkap dan sedetail aslinya (bumi).

### **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.


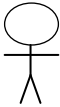

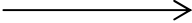
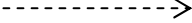

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

#### **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**




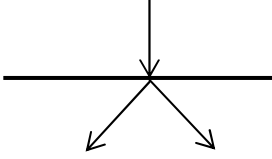
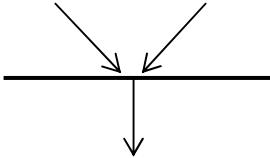
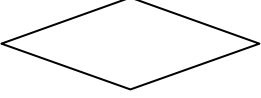

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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>.</p>
	<p>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.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(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* dapat dilihat pada Tabel II.2, yaitu:

**Tabel II.2. Simbol *Activity Diagram***

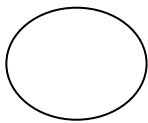
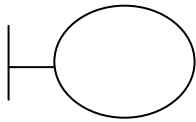
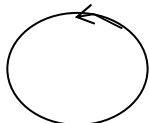
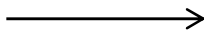
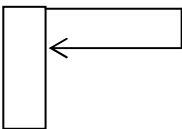


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel 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 diagram</i> 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.4. 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.3. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
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)