

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Informasi Geografis**

SIG merupakan salah satu sistem informasi. SIG merupakan suatu sistem yang menekankan pada unsur informasi geografi. Istilah “geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau tertukar hingga timbul istilah yang ketiga, geospasial. Ketiga istilah ini mengandung pengertian yang sama di dalam konteks SIG. Penggunaan kata “geografis” mengandung pengertian suatu persoalan mengenai bumi: permukaan dua atau tiga dimensi. Istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui (Koko Mukti Wibowo: 2015; 52).

#### **II.2. Konsep Dasar Web**

Web merupakan salah satu sumber daya internet yang berkembang pesat. Pendistribusian informasi web dilakukan melalui pendekatan *hyperlink*, yang memungkinkan suatu teks, gambar, ataupun objek yang lain menjadi acuan untuk membuka halaman-halaman yang lain. Melalui pendekatan ini, seseorang dapat memperoleh informasi dengan beranjak dari satu halaman ke halaman lain (Harison, Ahmad Syarif: 2016; 41).

### **II.3. SIG Berbasis Web**

Sistem Informasi Geografis telah berkembang dari segi keragaman aplikasi dan juga media. Pengembangan aplikasi SIG kedepannya mengarah kepada aplikasi berbasis *Web* yang dikenal dengan *Web SIG*. Hal ini disebabkan karena pengembangan aplikasi di lingkungan jaringan telah menunjukkan potensi yang besar dalam kaitannya dengan informasi geografis. Sebagai contoh adalah adanya peta *online* interaktif sebuah kota, yang memudahkan pengguna dalam mencari informasi geografis terkini yang terdapat pada kota tersebut, tanpa mengenal batas lokasi geografis pengguna. Pada aplikasi SIG berbasis *web*, terdapat beberapa komponen yang saling berinteraksi. Komponen-komponen tersebut bisa saja terdapat pada beberapa lokasi pada jaringan. Oleh karena itu pada SIG berbasis *web*, diperlukan adanya *server* (Harison, Ahmad Syarif: 2016; 41).

### **II.4. Metode Brown Gibson**

Metode Brown Gibson biasa digunakan untuk membantu analisis data dalam proses pengambilan keputusan yang memiliki multi atribut. Proses penilaian kandidat lokasi dengan menggunakan metode Brown Gibson akan menggunakan sistem bobot, di mana pada akhir penilaian kandidat lokasi yang memperoleh penilaian terbaik akan menjadi pilihan alternatif terbaik. Langkah-langkah untuk mengaplikasikan metode Brown Gibson dapat diuraikan sebagai berikut :

1. Mengeliminasi setiap alternatif pilihan yang secara sepintas jelas tidak layak dan layak untuk dipilih, atas dasar pertimbangan-pertimbangan teknis, atau keperluan lainnya dalam kapasitas alternatif yang dibutuhkan, dan bias dijadikan alasan utama untuk mengeliminasi suatu alternatif dalam daftar nominasi alternatif.
2. Menghitung dan tetapkan *performance measurement* dari Faktor Objektif (*OFi*) untuk setiap alternatif. Ukuran *performance* untuk Faktor Objektif dihitung berdasarkan estimasi seluruh perkiraan total biaya-biaya yang dikeluarkan untuk pemilihan alternative yang dipertimbangkan.

$$OFi = [Ci \cdot (1/Ci)]^{-1} \dots\dots\dots 2.2.1$$

3. Menentukan faktor-faktor yang memberi pengaruh signifikan dan harus dipertimbangkan pada saat pemilihan alternatif. Faktor-faktor ini lebih bersifat subjektif. Estimasi dari ukuran Faktor Subjektif (*SFi*) untuk setiap alternative pilihan ditentukan dengan menggunakan

$$SFi = Wj \cdot Rij \dots\dots\dots 2.2.2$$

4. Membuat pembobotan, mana yang lebih baik dipertimbangkan, antara Faktor Objektif (bobot =  $k$ ) dengan Faktor Subjektif (bobot =  $1 - k$ ) dari nilai batas(0, 1). Kombinasikan Faktor Objektif (*OFi*) dengan Faktor Subjektif (*SFi*) yang akan menghasilkan "*location preference measure*" (*LPMi*) untuk setiap alternatif yang ada.

$$LPMi = k (OFi) + (1 - k) (SFi) \dots\dots\dots 2.2.3$$

(Riska Ayu Pratiwi: 2016; 127).

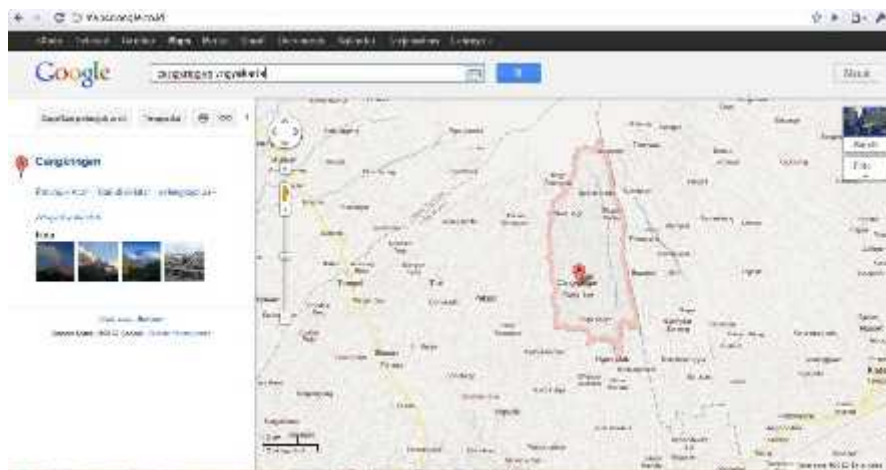
## II.5. Google Maps API

Google Map *Service* adalah sebuah jasa peta global virtual gratis dan online yang disediakan oleh perusahaan Google. Google Maps yang dapat ditemukan di alamat <http://maps.google.com>. Google Maps menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. Google Maps juga menawarkan pencarian suatu tempat dan rute perjalanan.

*Google Maps API* adalah sebuah layanan (*service*) yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam websitenya.

(Faya Mahdia: 2013; 164).

Adapun tampilan Google Maps API dapat dilihat pada Gambar II.1 :



**Gambar II.1. Tampilan Google Maps API**

## II.6. Pengertian PHP

*PHP* merupakan kependekan dari *Personal Home Page* (Situs Personal). *PHP* pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1995. Pada waktu itu, *PHP* masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengelola data formulir dari *web*. *PHP* disebut bahasa pemrograman *server-side*, karena *PHP* di proses pada komputer *server* (Agus Hariyanto: 2017; 12).

## II.7. Pengertian Database

Basis data atau *database* adalah kumpulan informasi yang disimpan di komputer secara sistematis dalam bentuk tabel- tabel, sehingga dapat diolah untuk memperoleh informasi dari *record yang tersimpan* dalam kolom-kolom pada tabel. Keberadaan *database* dalam sebuah aplikasi sangat penting, dimana dengan adanya *database* ini akan memungkinkan pengolahan data lebih dinamis (Agus Hariyanto: 2017; 62).

## II.8. Pengertian MySQL

*MySQL* merupakan salah satu *database* yang saat ini paling populer digunakan dalam lingkungan pengembangan aplikasi berbasis *web*. Selain kehadirannya, *MySQL* juga merupakan *database* relasional yang memiliki kecepatan akses data yang cukup tinggi. Oleh karena *database MySQL* dan *PHP* bagaikan dua sisi mata uang yang tidak bias dipisahkan (Agus Hariyanto: 2017; 65).

Adapun tampilan *MySql* dapat dilihat pada Gambar II.2.



**Gambar II.2 : Tampilan *MySQL***

## II.9. *Unified Modeling Language (UML)*

Menurut Windu Gata, Grace (2013: 4), *Unified Modeling Language (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 (Kusnita Yusmiarti: 2016; 3).


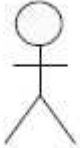


### II.9.1. *Use Case Diagram*

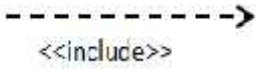
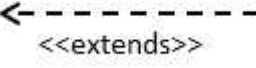
*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

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 dapat dilihat pada Tabel II.1 :

**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 aktir, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktir, 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 kontrol 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 mengindikasikan data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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>



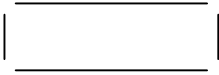
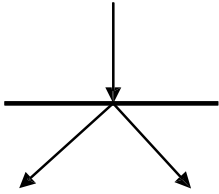
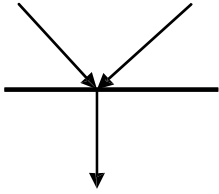

(Sumber : Kusnita Yusmiarti: 2016; 3)

### II.9.2. 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 :

Tabel II.2. Simbol *Activity Diagram*



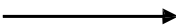
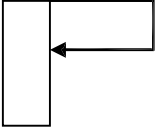


Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas.
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percaban gan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabung kan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabung an) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Kusnita Yusmiarti: 2016; 4)

### **II.9.3. *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* dapat dilihat pada Tabel II.3 :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</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>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form 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> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Kusnita Yusmiarti : 2016 : 5)

#### II.9.4. *Class Diagram*

*Class Diagram* 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 (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

**Tabel II.4. Simbol *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 maksimal 4

(Sumber : Kusnita Yusmiarti: 2016; 4)