

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis

SIG merupakan salah satu system informasi. SIG merupakan suatu sistem yang menekankan pada unsure 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 Euclidean Distance

Euclidean distance adalah metrika yang sering digunakan untuk menghitung kesamaan dua vektor. *Euclidean distance* menghitung akar dari kuadrat perbedaan dua vektor. Semakin besar jarak antara dua vektor, maka tingkat kesamaan atau kemiripannya kecil. Sebaliknya, semakin kecil jarak antara

dua vektor, maka tingkat kesamaan atau kemiripannya besar.

Rumus Euclidean distance adalah sebagai berikut :

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \dots\dots\dots (1)$$

(Sylvia Pretty Tulus ; 2014 : 104).

II.5. Quantum GIS

Quantum GIS merupakan salah satu perangkat lunak *open source* di bawah proyek resmi dari *Open Source Geospatial Foundation* (OSGeo) yang dapat dijalankan dalam system operasi *Windows, Mac OSX, Linux* dan *Unix*. Aplikasi ini menawarkan pengolahan data geospasial dengan berbagai format dan fungsionalitas vektor, raster dan *database*. Untuk keperluan analisis spasial, aplikasi ini telah cukup lengkap karena telah terintegrasi dengan perangkat lunak GRASS. Pemanfaatan perangkat lunak Quantum GIS ini dapat digunakan sebagai pilihan alternative dari software SIG komersial seperti *ArcView* maupun *ArcGIS*. Quantum GIS dapat diakses melalui situs resmi yang beralamatkan **Error! Hyperlink reference not valid.** Hussein, Werdiningsih ; 2012 : 93).

II.6. 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 computer *server* (Agus Hariyanto ; 2017 12).

II.7. Database

Basis data atau *database* adalah kumpulan informasi yang disimpan di computer secara sistematis dalam bentuk tabel- tabel, sehingga dapat di olah 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. 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 bisa dipisahkan (Agus Hariyanto ; 2017 :65).

II.9. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan system

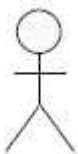
berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Kusnita Yusmiarti : 2016 : 3).

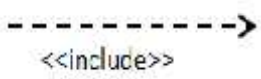
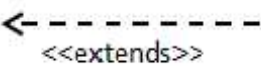
II.9.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan system informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam system 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
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan system sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja
	<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 system bias muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara actor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara actor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila actor 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.

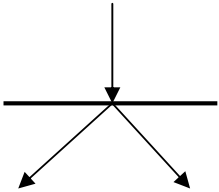
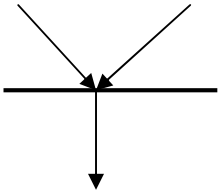
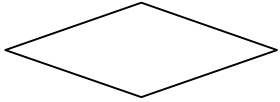
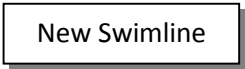
(Sumber :KusnitaYusmiarti : 2016 : 3)

II.9.2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis.

Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada TabelII.2 :

Tabel II.2. Simbol Activity Diagram

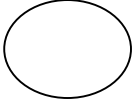
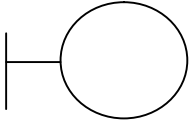
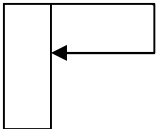
Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan 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 :KusnitaYusmiarti : 2016 : 4)

II.9.3. 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* 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 system 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 actor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
	<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> , symbol 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 kota kini 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 :KusnitaYusmiarti : 2016 : 5)

II.9.4. Diagram Kelas(*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggungjawab 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 kelas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* 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. *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 maksimal 4

(Sumber :Kusnita Yusmiarti : 2016 : 4)