

BAB II

LANDASAN TEORI

II.1. Sistem Informasi Geografis

Menurut Syaifudin Ramadhani, dkk (2013 : 480) Istilah geografi digunakan karena Sistem Informasi Geografis (SIG) dibangun berdasarkan pada geografi atau spasial. Objek ini mengarah pada spesifikasi pada lokasi dalam suatu space. *Geographic Information System* (GIS) merupakan sistem komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisis terhadap permukaan bumi.

Geografi adalah informasi mengenal permukaan bumi dan semua obyek yang berada di atasnya, sedangkan sistem informasi geografis (SIG) atau dalam bahasa Inggris disebut *Geographic Information System* (GIS) adalah sistem informasi khusus yang mengolah data yang memiliki informasi spasial (bereferensi keruangan). Sistem informasi geografis adalah bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antar muka. SIG tersusun atas konsep beberapa lapisan (layer) dan relasi.

II.1.1. Komponen Sistem Informasi Geografis

Menurut Bramantiyo Marjuki (2014 : 2) Sebagai salah satu jenis sistem informasi, SIG mempunyai sub sistem atau komponen yang bekerja secara

bersama untuk menghasilkan fungsionalitas SIG. Komponen SIG terdiri dari *hardware*, *software*, data dan metode.

1. *Hardware* atau perangkat keras merupakan media tempat pelaksanaan proses SIG. *Hardware* yang diperlukan dalam sebuah SIG meliputi perangkat keras untuk masukan data, penyimpanan data, pengolahan dan analisa data dan pembuatan keluaran.
2. *Software* atau perangkat lunak merupakan alat pelaksana pekerjaan SIG. *Software* standar SIG harus mempunyai kapabilitas data input, penyimpanan, manajemen data, transformasi dan konversi data, analisa dan pembuatan keluaran.
3. Data atau representasi dari sebuah obyek/ fenomena adalah bahan yang dianalisa di dalam SIG. SIG memerlukan sebuah jenis data yang spesifik agar dapat memberikan keluaran seperti fungsionalitasnya. Data yang digunakan dalam SIG adalah data geospasial atau data yang berefrensi geografis (mempunyai informasi lokasi).
4. Metode adalah cara bagaimana data diolah untuk menjadi sebuah informasi. Metode meliputi aspek pemasukan data ke dalam sistem, bagaimana data dikelola dan disimpan, bagaimana data dianalisis, dan bagaimana informasi ditampilkan

II.2. Metode *Euclidean Distance*

Euclidean Distance adalah metrika yang paling sering digunakan untuk menghitung kesamaan dua vektor. Rumus *euclidean Distance* adalah akar dari kuadrat perbedaan 2 vektor (*root of square differences between 2 vectors*)

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

Keterangan :

d_{ij} = tingkat perbedaan (*dissimilarity degree*)

n = jumlah vektor

x_{ik} = vektor citra input

x_{jk} = vektor citra pembanding /*output* (Sendhy Rachmat Wurdianarto ; 2014 : 35)

II.2.1. Studi Kasus Metode *Euclidean Distance*

$$d(x,y) = \|x-y\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

Pada penelitian ini, sebuah citra inputan yang akan dicari kesamaannya dengan sejumlah *template*, salah satu algoritma yang digunakan ialah *Euclidean distance*. Jika citra inputan sebagai $f(x,y)$ sedangkan sebuah citra *template* adalah $f(x,y)$ maka untuk menghitung beberapa nilai jarak antara dua citra tersebut yang dapat juga disebut sebagai nilai *error* adalah:

$$e(x,y) = f^i(x,y) - f(x,y)$$

Pada kasus pencocokan pola ini, ukuran matriks citra *template* dan citra masukan adalah sama. Jika ukuran matriks tersebut adalah $M \times N$ maka nilai *error* antara citra masukan dan citra *template* adalah:

$$e(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x,y) - f(x,y)]$$

Yang digunakan dalam penelitian ini adalah menghitung jarak antara rata-rata antara citra masukan dan citra keluaran kemudian dicari akar kuadrat nilai rata-rata dengan menggunakan persamaan :

$$e(x,y) = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x,y) - f(x,y)]^2 \right]^{\frac{1}{2}}$$

Untuk selanjutnya nilai tersebut disebut sebagai kolerasi. Kesamaan citra inputan dan citra *template* dapat dihitung dari nilai korelasi. Semakin kecil nilai kolerasi, maka kedua citra tersebut makin sama bentuknya. Pada penelitian ini di cari nilai korelasi yang paling kecil untuk menggantikan citra inputan dengan citra templete yang berupa karakter ASCII. (Febriliyan Samopa dkk ; 2002 : 35)

II.3. Pengertian *Hypertext Preprocessor* (PHP)

Hypertext Preprocessor (PHP) adalah bahasa pemrograman script yang paling banyak dipakai saat ini . PHP banyak di pakai untuk memrogram situs web dinamis walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh dari aplikasi PHP adalah forum (phpBB) dan mediaWiki (software dibelakan Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun

Microsystem s dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.(Syarifudin Ramadhani DKK ; 2013 : 480)

II.4. Pengertian Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya tersimpan diperangkat keras komputer dan digunakan sebagai perangkat lunak untuk memanipulasinya penerapan basis data (*database*) dalam sistem informasi disebut juga dengan *database system* . (Erna Kharistiani, dkk ; 2013 : 715).

II.5. Pengertian MySQL

MySQL adalah *Relational Database Management System (RDBMS)* yang di distribusikan secara gratis dibawah lisensi GPL (*General Public License*).Dimana setiap orang bebas menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). (Syarifudin Ramadhani DKK ; 2013 : 480).

II.6. Pengetian Peta

Peta adalah gambaran sebagian atau seluruh muka bumi baik yang terletak diatas maupun dibawah permukaan bumi dan di sajikan pada bidang datar pada

skala dan proyeksi tertentu (secara matematis). (Erna Kharistiani, dkk ; 2013 : 715).

II.6. Pengertian *Open Street Map* (OSM)

Open Street Map (OSM) merupakan suatu layanan peta *online* yang dibuat dan dikembangkan untuk didistribusikan secara gratis. Proyek ini dikembangkan karena sebagian besar layanan peta *online* yang ada mempunyai hak cipta dan batasan penggunaan berbayar sehingga menghambat orang-orang untuk menggunakannya secara kreatif maupun produktif. (Benardi Atmadja dkk; 2013 :3).

II.7. UML (*Unified Modeling Language*)

Menurut Windu Gata (2013 : 4) 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.


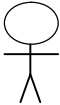

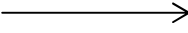
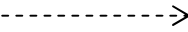

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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




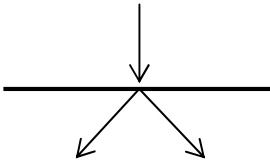
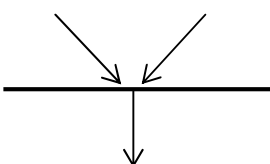
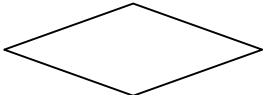
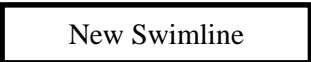
kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

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*

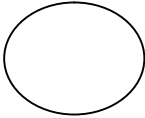
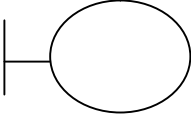
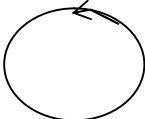

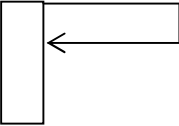

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau 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 Gata ; 2013 : 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>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>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 Gata ; 2013 : 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 Gata ; 2013 : 9)