BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis

GIS atau sistem informasi berbasis pemetaan dan geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang terkait dengan sistem pemetaan dan analisis terhadap segala sesuatu, serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi GIS mengintegrasikan operasi pengolahan data berbasis *database* yang biasa digunakan, seperti pengambilan data berdasarkan kebutuhan serta analisis statistic dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar tertentu.

Konsep GIS telah diperkenalkan di Indonesia sejak pertengahan tahun 1980-an, dan kini telah dimanfaatkan di berbagai bidang baik negeri maupun swasta. Kemampuan dasar dari GIS adalah mengintegrasikan berbagai operasi basis data seperti *quIery*, menganalisisnya, dan menyimpan serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan GIS dengan sistem informasi lain. Komponen GIS terdiri atas *hardware*, *software*, data, dan *user*. Dengan adanya GIS diharapkan tersedia informasi yang cepat, benar dan akurat tantang keadaan di lingkungannya (Hersa Farida Qoriani ; 2012 : 2).

II.1.1. *Data Spasial*

Menurut Mohd. Ichsan (2012 : 51), Sebagian besar data yang akan ditangani dalam SIG merupakan data spasial, data yang berorientasi geografis. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut) yang dijelaskan berikut ini:

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi nonspasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya. Contoh jenis vegetasi, populasi, luasan, kode pos, dan sebagainya.

II.1.2. *Format Data Spasial*

Menurut Mohd. Ichsan (2012 : 51), Secara sederhana format dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara file satu dengan lainnya. Dalam SIG, data spasial dapat direpresentasikan dalam dua format, yaitu:

a. Data vektor

   Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (titik perpotongan antara dua buah garis).

b. Data raster

Data raster (disebut juga dengan sel grid) adalah data yang dihasilkan dari sistem penginderaan jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan *pixe*l (*picture element*).

II.1.3. Data Vektor

Data Vektor merupakan bentuk bumi yang dipresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (merupakan titik perpotongan antara dua buah garis). Keuntuungan utama dari format data vektor adalah ketepatan dalam merepresentasikan fitur titik, batasan dan garis lurus. Hal ini sangat berguna untuk analisa yang membutuhkan kepetapan posisi, misalnya pada basisdata batas-batas kedaster. Contoh pengguna lainnya adalah untuk mendefinisikan hubungan spasial dari beberapa fitur. Kelemahan data vektor yang utama adalah ketidakmampuannya dalam mengakomodasi perubahan gradual (Bramantiyo Marjuki ; 2014 : 5).

II.1.4. Data Raster

Data raster (atau disebut juga dengan sei grid) adalah data yang dihasilkan dari sistem pengideraan jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan pixel (*picture element*). Pada data raster, resolusi (definisi visual) tergantung pada ukuran pikselnya. Dengan kata lain, resolusi piksel menggambarkan ukuran sebenarnya di permukaa bumi yang

diwakili oleh setiap piksel pada citra. Semakin kecil ukuran permukaan bumi yang direpresentasikan oleh satu sel, semakin tinggi resolusinya. Data raster sangat baik untuk merepresentasikan batas-batas yang berubah secara gradual, seperti jenis tanah, kelembaban tanah, vegetasi, suhu tanah dan sebagainya. Keterbatasan utama dari data raster adalah besarnya ukuran file, semakin tinggi resolusi gridnya semakin besar pula ukuran filenya dan sangat tergantung pada kapasitas perangkat keras yang tersedia (Bramantiyo Marjuki ; 2014 : 5).

II.2. Laundry

Laundry adalah salah satu kegiatan rumah tangga yang menggunakan deterjen sebagai bahan pembantu untuk membersihkan pakaian, karpet, dan alat-alat rumah tangga lainnya. Kehadiranjasa laundry ini dapat membawa manfaat yang cukup besar bagi perekonomian dengan megurangi jumlah pengangguran serta meningkatkan taraf hidup manusia. Namun limbah laundry juga dapat menimbulkan pencemaran lingkungan terutama adanya deterjen, jika limbah yang dihasilkan tidak diolah terlebih dahulu sebelum dibuang. Deterjen mengandung zat surface active (surfaktan), yaitu anionik, kationik, dan nonionik. Surfaktan yang digunakan dalam deterjen adalah jenis anionik dalam bentuk sulfat dan sulfonat. Surfaktan sulfonat yang dipergunakan adalah Alkyl Benzene Sulfonate (ABS) dan Linier Alkyl Sulfonate (LAS). Lingkungan perairan yang tercemar limbah deterjen kategori keras ini dalam konsentrasi tinggi dapat membahayakan kehidupan biota air dan manusia yang mengkonsumsi biota tersebut (Yuli Pratiwi ; 2012 : 298).

II.3. Metode Haversine

Menurut Dwi Prasetyo (2012 : 2), Metode Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*lattitude*) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinant bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut :

x = (lon2-lon1)*cos((lat1+lat2)/2);

y = (lat2-lat1);

d = sqrt(x*x+y*y)*R

Keterangan :

x = Longitude (Lintang)

y= Lattitude (Bujur)

d= Jarak

R= Radius Bumi =6371 km 1 derajat= 0.0174532925 radian.

II.4. PHP

PHP merupakan suatu bahasa pemrograman sisi server yang dapat anda gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page (*ASP) dan *Java Server Page*(JSP). Dalam suatu

halaman HTML anda dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi. Karena kekayaannya akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki popularitas yang tinggi. Anda dapat mengecek *survey* popularitas yang dilakukan *netcraft* di URL www.php.net/usage.php.

PHP adalah kependekan dari *HyperText Prepocessor* (suatu akronim rekursif) yang dibangun oleh RasmusLerdorf pada tahun 1994. Dahulu, pada awal pengembangnnya PHP disebut sebagai kependekan dari *Personal Home Page.* PHP merupakan produk *Open Source*s sehingga anda dapat mengakses *source code*, menggunakan dan mengubahnya tanpa harus membayar sepeser pun. Gratis! (Antonius Nugraha Widhi Pratama ; 2010 : 9).

II.5. Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk.* Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan.

Misalnya, kita bisa menggunakan mesin ATM *(*anjungan tunai mandiri / *automatic teller machine*) bank karena banktelah mempunyai database tentang nasabah dan rekening nasabah.Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan *(update)* data pada *database*di bank.Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir*(form)* yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya (Agustinus Mujilan ; 2012 : 23).

II.6. MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle, Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan mengambil data secara efesien. *Server* MySQL yang akan membantu melakukan fungsionaliitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah SQL-*standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial.

Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web. Anda tidak perlu membayar, akan tetapi jika ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian mengualnua, anda wajib bertemu mudah dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).

Gambar II.1. Tampilan MySQL Server
(Sumber : Antonius Nugraha Widhi Pratama ; 2010 : 10)

II.7. Teknik Normalisasi

Menurut Janner Simarmata (2010 : 79-84), Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang tekah ditemukan

II.7.1.Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

   Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1" Normal Form)

   Definisi :

   Sebuah table disebut 1NF jika :

   - Tidak ada baris yang duplikat dalam tabel tersebut.
   - Masing-masing cell bernilai tunggal

   Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

c. Bentuk normal tahap kedua (2^{nd} normal form)

   Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

d. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi X -> A, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey.

II.8. 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.

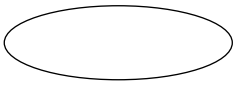
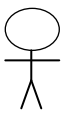
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

1. *Use case* Diagram

   Use case diagram merupakan pemodelan untuk kelakukan (*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
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.

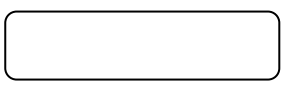
	Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara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

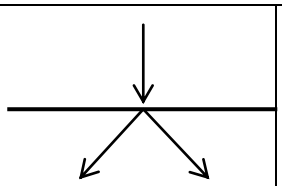
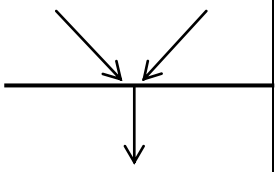
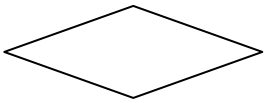
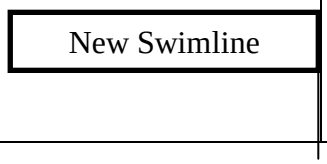
(Sumber : Windu Gata ; 2013 : 4)

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
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.

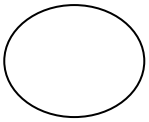
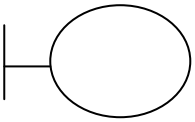
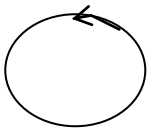
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false*.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

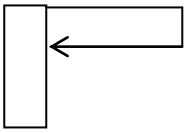
(Sumber : Windu Gata ; 2013 : 6)

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
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	*Message*, simbol mengirim pesan antar *class*.
	*Recursive,* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation, activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(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 tanggng 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)