

BAB II

LANDASAN TEORI

II.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.(Sulindawati ; 2010 : 135)

II.2. Pengertian Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimaannya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi. Kualitas dari suatu informasi tergantung pada tiga hal yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan tersebut, maka dapat berakibat fatal untuk organisasi.

Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. (Sulindawati ; 2010 : 2).

II.3. Sistem Informasi Geografis

SIG sebagai suatu sistem yang mengorganisir *hardware*, *software*, dan data, serta dapat mendayagunakan sistem penyimpanan, pengolahan, maupun analisis data secara simultan, sehingga dapat diperoleh informasi yang berkaitan dengan aspek keruangan. Selanjutnya Purwadhi (1994) juga menjelaskan bahwa SIG merupakan manajemen data spasial dan non-spasial yang berbasis komputer dengan tiga karakteristik dasar, yaitu: mempunyai fenomena aktual (variabel data nonlokasi) yang berhubungan dengan topik permasalahan di lokasi bersangkutan, merupakan suatu kejadian di suatu lokasi dan mempunyai dimensi waktu. Istilah SIG juga didefinisikan sebagai sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Dalam arti yang lebih sempit, SIG adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut

lokasinya, dalam sebuah *database*. Para praktisi juga memasukkan orang yang membangun dan mengoperasikan SIG dan data sebagai bagian dari sistem ini. Teknologi SIG dapat digunakan untuk berbagai kepentingan, seperti investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute (Edhy Sutanta ; 2010 : 38)

II.3.1. Komponen Sistem Informasi Geografis

SIG merupakan sistem kompleks yang terintegrasi dengan sistem-sistem computer lain di tingkat fungsional dan jaringan. Sistem SIG terdiri dari beberapa komponen berikut :

1. Perangkat keras

Pada saat ini perangkat SIG dapat digunakan dalam berbagai platform perangkat keras mulai dari *PC Desktop*, *workstation* hingga *multi user host* yang digunakan oleh banyak orang secara bersamaan dalam jaringan luas. Perangkat keras yang sering digunakan untuk SIG adalah komputer (*PC*), *mouse*, *digitizer*, *printer*, *plotter* dan *scanner*.

2. Perangkat lunak

SIG juga merupakan sistem perangkat lunak yang tersusun secara modular dimana basis data memegang peranan kunci. Setiap sub-sistem diimplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul, hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang masing-masing dapat dieksekusi sendiri. (Nurpilihan Bafdal dkk: 2011 ; 7)

II.3.2. Data Spasial

Data spasial mempunyai pengertian sebagai suatu data yang mengacu pada posisi, obyek, dan hubungan diantaranya dalam ruang bumi. Data spasial merupakan salah satu item dari

informasi, dimana didalamnya terdapat informasi mengenai bumi termasuk permukaan bumi, dibawah permukaan bumi, perairan, kelautan dan bawah atmosfer (Rajabidfard dan Williamson, 2000). Data spasial dan informasi turunannya digunakan untuk menentukan posisi dari identifikasi suatu elemen di permukaan bumi (Radjabidfard 2001). Lebih lanjut lagi Mapping Science Committee (1995) dalam Rajabidfard (2001) menerangkan mengenai pentingnya peranan posisi lokasi yaitu, :

- (1) pengetahuan mengenai lokasi dari suatu aktifitas memungkinkan hubungannya dengan aktifitas lain atau elemen lain dalam daerah yang sama atau lokasi yang berdekatan .
- (2) Lokasi memungkinkan diperhitungkannya jarak, pembuatan peta, memberikan arahan dalam membuat keputusan spasial yang bersifat kompleks. (Nurpilihan dkk : 2011 ; 19)

II.3.3. *Format Data Spasial*

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 *pixel* (*picture element*). (Nurpilihan Bafdal dkk; 2011 : 18)

II.3.4. Model Data Vektor

Model data vektor merupakan model data yang paling banyak digunakan, model ini berbasiskan pada titik (*point*) dengan nilai koordinat (x,y) untuk membangun obyek spasialnya. Obyek yang dibangun terbagi menjadi tiga bagian lagi yaitu berupa titik (*point*), garis (*line*), dan area (*polygon*).

- **Titik (point)** Titik merupakan representasi grafis yang paling sederhana pada suatu obyek. Titik tidak mempunyai dimensi tetapi dapat ditampilkan dalam bentuk simbol baik pada peta maupun dalam layar monitor. Contoh : Lokasi Fasilitas Kesehatan, Lokasi Fasilitas Kesehatan.
- **Garis (line)** Garis merupakan bentuk linear yang menghubungkan dua atau lebih titik dan merepresentasikan obyek dalam satu dimensi. Contoh : Jalan dan Sungai.
- **Area (Poligon)** Poligon merupakan representasi obyek dalam dua dimensi. Contoh : Danau dan Persil Tanah. (Nurpilihan Bafdal; 2011 : 30).

II.3.5. Model Data Raster

Model data *raster* mempunyai struktur data yang tersusun dalam bentuk matriks atau piksel dan membentuk *grid*. Setiap piksel memiliki nilai tertentu dan memiliki atribut tersendiri, termasuk nilai koordinat yang unik. Tingkat keakurasian model ini sangat tergantung pada ukuran piksel atau biasa disebut dengan *resolusi*. Model data ini biasanya digunakan dalam remote sensing yang berbasiskan citra satelit maupun *airborne* (pesawat terbang). Selain itu

model ini digunakan pula dalam membangun model ketinggian digital (DEM-Digital Elevation Model) dan model permukaan digital (DTM-Digital Terrain Model). Model raster memberikan informasi spasial terhadap permukaan di bumi dalam bentuk gambaran yang di generalisasi. dunia nyata disajikan sebagai elemen matriks atau piksel yang membentuk grid yang homogen. Pada setiap piksel mewakili setiap obyek yang terekam dan ditandai dengan nilai-nilai tertentu. Secara konseptual, model data raster merupakan model data spasial yang paling sederhana. (Nurpilihan Bafdal ; 2011 : 22).

II.4. *Android*

Android merupakan generasi baru *platform mobile*, platform yang memberikan pengembang untuk melakukan pengembangan sesuai yang diharapkannya. Sistem operasi yang mendasari *android* dilisensikan di bawah GNU, *general public lisensi* versi 2 (GPLv2), yang lebih dikenal dengan istilah *copyleft*, lisensi di mana setiap perbaikan pihak ketiga harus terus dibawah syarat (terms). *Android* di distribusikan di bawah *Lisensi Apache Software* (ASL/Apache 2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan platform tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi *android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Adnyana dan Efendi ; 2014 : 10).

II.5. *Google Maps*

Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di <http://maps.google.com>. Google Maps menyediakan gambar resolusi tinggi satelit untuk daerah perkotaan sebagian besar di Amerika Serikat (termasuk Hawaii, Alaska, PuertoRico, dan US Virgin Islands), Kanada, dan Inggris, serta sebagai bagian dari Australia dan banyak negara lainnya. Seperti banyak aplikasi web lainnya Google, Google Maps menggunakanJavaScript secara ekstensif. Beberapa tujuan dari penggunaan Google Maps API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan Google Maps. (Nurul Azmi : 2013; 5).

Adapun tampilan *Google Maps* dapat dilihat pada gambar II.1 dibawah ini :



Gambar II.1. Tampilan *Google Maps*

II.6. *MySQL*

MySQL (bisa dibaca dengan mai-es-ki-el atau bisa juga mai-se-kuel) adalah suatu perangkat lunak database relasi (*Relational Database Management System* atau *DBMS*), seperti halnya ORACLE, POSTGRESQL, MSSQL, dan sebagainya. SQL merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau

bahasa program yang digunakan untuk mengelola suatu database. Jadi MySQL adalah software-nya dan SQL adalah bahasa perintahnya. (Aniysa 2013:51)

Adapun tampilan mysql dapat dilihat pada gambar II.2 dibawah ini :



Gambar II.2. Tampilan MySQL

II.7. Eclipse

Eclipse merupakan sebuah editor, secara default editor ini belum bisa dipakai untuk mendevlop android. Agar bisa digunakan untuk membuat aplikasi android maka harus diinstall dahulu plugin, namanya ADT (*Android Development Tools*). Setelah terinstal, maka eclipse sudah siap digunakan, hanya saja belum dilengkapi library dan emulator. Untuk memenuhinya, maka perlu diinstal SDK *Android (StKamurt Development Kit)*. Secara garis besar, eclipse bias digunakan setelah semua selesai mengkonfigurasi antara *editor Eclipse*, ADT dan SDK. Sebelumnya, terlebih dahulu harus memiliki 3 buah komponen dibawah ini. Dengan mengunduhnya pada situs penyedia layanan android. (Saputro: 2011 : 6)

II.8. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang digunakan sebagai pepadu 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 relasional.

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 telah ditemukan.

II.8.1. Bentuk-Bentuk Normalisasi

1. Bentuk Normal Tahap Pertama (1st Normal Form)

Contoh yang kita gunakan disini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk Normal Tahap Kedua (2nd Normal Form)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada INF, tetapi tidak pada 2NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada INF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk Normal Tahap Ketiga (3rd Normal Form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasioanl tergantung hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika

tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan *multivalue* (MDV).

6. Bentuk Normal Tahap Kelima

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sebuah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). (Janner Simarmata; 2010: 79)

II.10. 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.

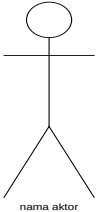
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem, dan merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan 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 *actor* 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 yang digunakan dalam *use case* diagram, yaitu sebagai berikut

Tabel II.1. Simbol-Simbol pada diagram Use Case

Simbol	Nama	Keterangan
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan actor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>

	Aktor	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.
	Asosiasi	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.
<<include>> 	Include	<i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pengambilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.

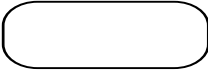
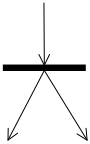
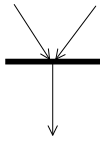
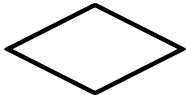
<<extend>> 	<i>Extend</i>	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.
---	---------------	---

(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 aktivitas.
	<i>End point</i> , akhir aktivitas.
	<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 menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Point</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk

New Swimlane	menunjukkan siapa melakukan apa.
--------------	----------------------------------

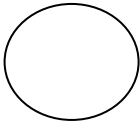
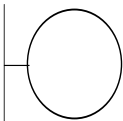
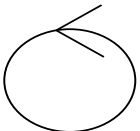
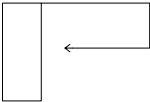
(Sumber : Windu Gata; 2013: 6)

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 actor 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> , symbol mengirim pesan antar class.
	<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 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*

<i>Multiplicity</i>	<i>Penjelasan</i>
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)