

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1 Sistem**

Menurut Hanif Al Fatta (2007 : 3), Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. Murdick dan Ross (1993) mendefinisikan sistem sebagai perangkat elemen yang digabungkan satu sama lainnya untuk suatu tujuan bersama.

Dari defenisi di atas, dapat diambil kesimpulan mengenai pengertian sistem yaitu sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan, atau dapat pula disimpulkan sebagai berikut sistem yaitu satu kesatuan yang saling berinteraksi atau terkait yang bergantung satu sama lainnya. Untuk mencapai suatu tujuan yang keseluruhannya memiliki *input*, *process*, *output*, dan *feedback*.

Menurut Kusrini (2007 : 11) “Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*)”.

Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan stategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar.

Menurut Kusrini (2007: 11) sistem informasi dibagi menjadi:

1. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau TPS)
2. Sistem Informasi Manajemen (*Management Information System* atau MIS)
3. Sistem Otomasi Perkantoran (*Office Automation System* atau OAS)
4. Sistem Pendukung Keputusan (*Decision Support System* atau DSS)
5. Sistem Informasi Eksekutif (*Executive Information System* atau EIS)
6. Sistem Pendukung Kelompok (*Group Support System* atau GSS)
7. Sistem Pendukung Cerdas (*Intelligent Support System* atau ISS)

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk mendukung manajemen, maka ketiga sistem tersebut sering disebut sistem pendukung manajemen (*management support system* atau MSS).

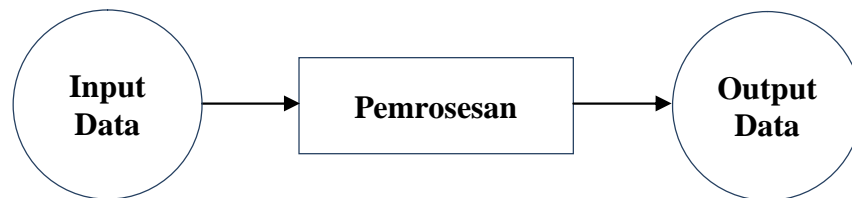
## **II.2 Informasi**

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai, sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

## **II.3 Sistem Informasi**

Menurut Kertahadi (Hanif Al Fatta ; 2007 : 9) Sistem Informasi Manajemen (SIM) dapat didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi

subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada proses. Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output – IPO*) dapat dilihat pada gambar berikut :



**Gambar II.1. Konsep Sistem Informasi**  
( Sumber : Hanif Al Fatta ; 2007 )

#### **II.4 Sistem Informasi Geografis**

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. Dengan SIG kita akan dimudahkan dalam melihat fenomena kebumih dengan perspektif yang lebih baik. SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta analog bahkan data statistik.

Sejak pertengahan tahun 1970-an, telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang bereferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup :

1. Pengorganisasian data dan informasi.
2. Penempatan informasi pada lokasi tertentu.
3. Melakukan komputasi, memberikan ilustrasi keterhubungan satu sama lainnya (koneksi), beserta analisis-analisis spasial lainnya.

Sebutan umum untuk sistem-sistem yang menangani masalah-masalah di atas adalah Sistem Informasi Geografis (SIG). Sistem Informasi Geografis adalah sistem komputer yang digunakan untuk memasukkan, menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang berhubungan dengan posisi-posisi dipermukaan bumi (Dewi ; 2010 : 1).

Pada awalnya, data geografis hanya disajikan di atas peta dengan menggunakan simbol, garis, dan warna. Elemen-elemen geometri ini dideskripsikan di dalam legendanya, misalnya garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan berikutnya. Selain itu berbagai data juga dapat di-*overlay*-kan berdasarkan sistem koordinat yang sama. Akibatnya sebuah peta menjadi media yang efektif baik sebagai alat presentasi maupun sebagai tempat penyimpanan data geografis ( Dewi ; 2010 : 2).

Sistem informasi merupakan suatu sistem yang tujuannya menghasilkan informasi. Sistem Informasi Geografis atau SIG merupakan sejenis perangkat lunak yang dapat digunakan untuk memasukkan, menyimpan, memanipulasi, dan menampilkan informasi geografis berikut atribut-atributnya.

Komponen SIG adalah sistem komputer, data geospasial, dan pengguna. SIG mengolah 2 macam data yaitu data geospasial atau yang biasanya disebut data spasial dan data nonspasial (atribut). Dalam SIG hal yang paling dipentingkan adalah tampilan data secara spasial, walaupun sebenarnya pada SIG kadang-kadang juga melibatkan data atribut, baik secara langsung maupun tidak (Suryani *et al* ; 2012 : 40).

## **II.5 Global Positioning System (GPS)**

*Global Positioning System (GPS)* adalah sistem berbasis satelit yang dapat digunakan dalam navigasi untuk mencari posisi di mana saja di bumi. GPS dirancang dan dioperasikan oleh *US Department of Defense (DOD)*. GPS terdiri dari satelit, kontrol dalam memantau stasiun dan GPS penerima. Penerima GPS mengambil informasi yang ditransmisikan dari satelit dan menggunakan triangulasi untuk menghitung lokasi pengguna yang tepat. Penggunaan GPS dapat digunakan untuk berbagai fungsi (Pooja Singal dan Chhillar ; 2014 : 12) :

1. Untuk menentukan posisi lokasi.
2. Untuk menavigasi dari satu lokasi ke lokasi lain.
3. Untuk membuat peta digital.
4. Untuk menentukan jarak antara dua titik.

### **II.5.1 Cara Kerja GPS**

Dasar GPS adalah satelit yang terus mengorbit mengelilingi bumi. Ini dilengkapi dengan jam atom dan sinyal radio pancar yang berisi tepat mereka lokasi, waktu dan informasi lainnya. Sinyal radio yang ditransmisikan dari satelit dipantau dan dikoreksi oleh stasiun kontrol yang dikirim kembali ke satelit menggunakan antena *ground*. Sinyal radio dari satelit dijemput oleh penerima GPS. Sebuah penerima GPS hanya membutuhkan 3 satelit untuk plot, posisi 2D kasar, yang tidak akan sangat akurat. Idealnya, 4 atau lebih satelit yang diperlukan

untuk merencanakan posisi 3D, yang lebih akurat daripada 2D (Pooja Singal dan Chhillar ; 2014 : 12).

### II.5.2. Tiga Segmen dari GPS

Berikut ini merupakan tiga segmen dari GPS (Pooja Singal dan Chhillar ; 2014 : 12) :

#### 1. Ruang Segmen

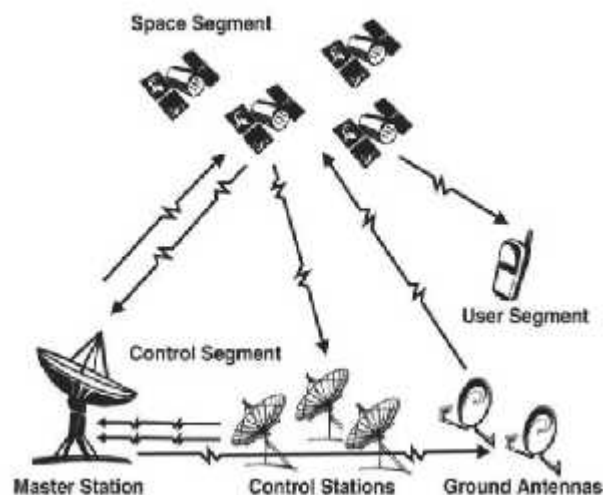
Ruas angkasa : satelit yang mengorbit mengelilingi bumi.

#### 2. Kontrol Segmen

Segmen kontrol : Kontrol & stasiun pemantauan.

#### 3. Pengguna Segmen

Segmen Pengguna : Penerima GPS yang dimiliki oleh warga sipil dan militer.



**Gambar II.2. Tiga Segmen Dari GPS**

*(Sumber : Pooja Singal dan Chhillar ; 2014)*

### II.5.3 Location Based Service (LBS)

Menurut Qusay dalam M.Rofiq (2014 : 51), *Location Based Service* adalah sebuah layanan yang digunakan untuk mengetahui posisi dari pengguna, kemudian menggunakan informasi tersebut untuk menyediakan jasa dan aplikasi yang personal.

Ada 2 pendekatan dasar yang dipakai dalam mengimplementasikan LBS, yaitu :

1. Memproses data lokasi di server dan mengirimkan hasilnya ke alat.
2. Mendapatkan data lokasi dari alat tersebut berdasarkan aplikasi yang menggunakannya secara langsung.

Aplikasi dapat menggunakan beberapa pendekatan yang disediakan, yaitu :

1. Menggunakan jaringan telepon seluler, akurasi dari metode ini tergantung kepada ukuran dari sel dan mungkin tidak akurat.
2. Menggunakan satelit, dengan menggunakan bantuan GPS untuk mendapatkan data posisi yang akurat.

### II.6 Metode Haversine

Metode *haversine* adalah suatu metode untuk menghitung jarak dari suatu tempat ke tempat tujuan. Proses kalkulasi jarak membutuhkan titik koordinat *latitude* dan *longitude* tempat asal serta titik koordinat *latitude* dan *longitude* tempat tujuan. Berikut ini rumus *haversine* (Sariyun Naja, 2012).

$$\text{lat} = \text{lat2} - \text{lat1}$$

$$\text{long} = \text{long2} - \text{long1}$$

$$a = \sin^2(\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\text{long}/2)$$

$$c = 2 \cdot \text{atan2}(v_a, v(1-a))$$

$$d = R \cdot c$$

Keterangan :

R = jari-jari bumi sebesar 6371(km)

lat = besaran perubahan *latitude*

long = besaran perubahan *longitude*

c = kalkulasi perpotongan sumbu

d = jarak (km)

## II.7 Website

*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi, gambar gerak, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan *link-link*. *Website* juga dapat diartikan sebagai kumpulan halaman-halaman yang digunakan mempublikasikan informasi berupa *teks*, gambaran dan program multimedia lainnya berupa (gambar gerak, tulisan gerak), suara ataupun gabungan sari semuanya, itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu page dengan page yang lain yang sering disebut *hyperlink*.

*Terminology website* adalah kumpulan dari halaman-halaman situs yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada didalam *Word Wide Web (WWW)* di *internet*. Sebuah *web page* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang

hampir selalu bias diakses melalui HTTP, yaitu *protocol* yang menyampaikan informasi dari server *website* untuk ditampilkan kepada para pemakai melalui *webbrowser* (Ahmad, 2011).



## II.8 Diagram Dasar dalam *Unified Modeling Language* (UML)


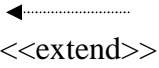
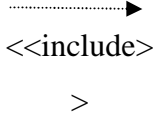
Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya:

### II.8.1. Model *Use Case Diagram*

*Use Case Diagram* secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *Use Case Diagram* secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use Case Diagram* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi (Sugiarti.Y, 2013: 37). Berikut adalah simbol-simbol yang ada pada *Use Case Diagram* :

**Tabel II.1. Simbol-Simbol *Use Case***

No	Gambar	Nama	Keterangan
1		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2	 Nama Actor	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi

			walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, miip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, arah panah menuntuk pada <i>use case</i> yang dituju.
5		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

(Sumber : Sugiarti.Y, 2013 : 37)

## II.8.2. Diagram Struktur Statis

UML menawarkan dua diagram untuk memodelkan struktur statis sistem informasi, yaitu (Sugiarti.Y, 2013: 37):

- a. *Class Diagram*, menggambarkan struktur *object* sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.
- b. *Object Diagram*, serupa dengan *class* diagram, tetapi *object* diagram memodelkan *instanceobject actual* dengan menunjukkan nilai-nilai saat ini dari atribut *instance*. *Object* diagram menyajikan "snapshot/potret" tentang objek sistem pada *point* waktu tertentu. Diagram ini tidak digunakan sesering *Class Diagram*, tetapi saat digunakan dapat membantu seorang *developer* memahami struktur sistem secara lebih baik.

### II.8.3 Diagram Interaksi

Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu (Sugiarti.Y, 2013 : 38).

#### 1. Diagram Rangkaian atau *Sequence Diagram*

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau *timing* apa.

#### 2. Diagram kolaborasi atau *Collaboration Diagram*

Serupa dengan diagram rangkaian atau sekuensi, tetapi tidak fokus pada *timing* atau sekuensi pesan. Diagram ini justru menggambarkan interaksi atau kolaborasi antara objek dalam sebuah format jaringan.

Diagram rangkaian maupun diagram kolaborasi merupakan isomorphic artinya dapat mengubah dari satu diagram ke diagram lain.

#### **II.8.4. Diagram *State* atau *State Diagram***

UML memiliki sebuah diagram untuk memodelkan behavior objek khusus yang kompleks (*statechart*) dan sebuah diagram untuk memodelkan behavior dari sebuah *use case* atau sebuah metode, yaitu (Sugiarti, 2013: 38):

1. *Diagram stechart*, digunakan untuk memodelkan *behavior* objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event* (kejadian) yang menyebabkan objek beralih dari satu *state* ke *state* lain.
2. Diagram aktivitas atau *Activity Diagram*, secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity* diagram dapat juga digunakan untuk memodelkan *action* tersebut.

#### **II.8.5. Diagram Implementasi**

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu (Sugiarti, 2013: 38).

1. Diagram komponen atau *Component Diagram*, digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. Komponen diagram dapat digunakan untuk menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (atau komponen).

2. Diagram penguraian atau *Deployment*, digunakan untuk mendeskripsikan arsitektur fisik dalam istilah “node” untuk *hardware* dan *software* dalam sistem. Diagram ini menggambarkan konfigurasi komponen-komponen *software real-time*, *processor*, dan peralatan yang membentuk arsitektur sistem.

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

- a. *Use case diagram*
- b. *Class diagram*
- c. *Statechart diagram*
- d. *Activity diagram*
- e. *Sequence diagram*
- f. *Collaboration diagram*
- g. *Component diagram*
- h. *Deployment diagram*

## **II.9 Database**

*Database* secara sederhana, dapat disebut sebagai gudang data. Secara teori, database adalah kumpulan data atau informasi yang kompleks, data-data tersebut disusun menjadi beberapa kelompok dengan tipe data yang sejenis (disebut tabel), di mana setiap datanya dapat saling berhubungan satu sama lain atau dapat berdiri sendiri, sehingga mudah diakses (Nugroho.B, 2008:1).

Dalam basis data, *database* dapat dikelola menggunakan program aplikasi basis data, misalnya : *MySQL*, *Access*, *PostgreSQL*, dan *SQL Server*. Aplikasi yang dapat membuat dan mengelola *database* disebut SMDB (Sistem Manajemen Basis Data) atau DBMS (*Database Manajemen System*).

Ada dua perbedaan SMDB yang digunakan, yaitu *database* yang bersifat *Stand Alone* dan *database* yang bersifat *Database Server*. Berikut ini, pengkategorianya:

1. *DatabaseServer* (berjalan *client* dan *server*)
  - a. *MySQL*
  - b. *PostgreSQL*
  - c. *SQL Server*
  - d. *Oracle*
  - e. *Interbase*
2. Bukan *DatabaseServer* (berjalan *Stand Alone*)
  - a. *Microsoft Access*
  - b. *Paradox*
  - c. *dBase*

## II.10 *MySQL*

*MySQL(Structured Query Language)* adalah *database server* yang awalnya hanya berjalan pada sistem Unix dan Linux. Seiring berjalannya waktu dan banyaknya peminat yang menggunakan *database* ini, *MySQL* merilis versi yang dapat diinstal pada hampir semua *platform*, termasuk *Windows*.

Lisensi dari *MySQL* adalah *freeware*. Yang dapat diunduh dan menggunakannya tanpa harus membayarnya. Meskipun kita menjual produk menyertakan *software MySQL*, tidak melanggar hak cipta.

Sedangkan *SQL (Structured Query Language)* merupakan suatu bahasa permintaan terstruktur. *SQL* memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. *SQL* merupakan bahasa permintaan yang melekat pada satu *database* atau *SMBD* tertentu (Nugroho. B, 2008: 2).

### II.10.1 Perintah *SQL*

Dalam penggunaannya, perintah *SQL* dikategorikan menjadi tiga sub perintah, yaitu (Nugroho. B, 2008: 5):

#### 1. *Data Definition Language (DDL)*

Merupakan sub bahasa *SQL* yang digunakan untuk membangun kerangka

*database*. Ada tiga perintah yang termasuk dalam *DDL*, yaitu :

##### a. *CREATE*

Perintah ini digunakan untuk membuat, termasuk membuat *database* baru, tabel baru, *view* baru, dan kolom.

##### b. *ALTER*

Perintah ini digunakan untuk mengubah struktur tabel yang telah dibuat.

Pekerjaan mencakup mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom, maupun memberikan atribut pada kolom.

##### c. *DROP*

Perintah ini digunakan untuk menghapus *database* dan tabel.

## 2. *Data Manipulation Language (DML)*

Merupakan sub bahasa *SQL* digunakan untuk memanipulasi data dalam database yang telah terbuat. Perintah yang digunakan, diantaranya :

### a. *INSERT*

Perintah ini digunakan untuk menyisipkan atau memasukan data baru ke dalam tabel.

### b. *SELECT*

Perintah ini digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel dalam relasi.

### c. *UPDATE*

Perintah ini digunakan untuk memperbarui data lama menjadi data terkini.

### d. *DELETE*

Perintah ini digunakan untuk menghapus data dari tabel.

## 3. *Data Control Language (DCL)*

Merupakan sub bahasa *SQL* yang digunakan untuk melakukan pengontrolan data dan *server database*, diantaranya:

### a. *GRANT*

Perintah ini digunakan untuk memberikan hak atau izin akses oleh administrator *server* kepada *user*.

### b. *REVOKE*

Perintah ini memiliki kegunaan terbalik dengan *GRANT*.