

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem merupakan suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu (Tata Sutabri, 2012 : 10). Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*). Di samping itu, suatu sistem senantiasa tidak lepas dari lingkungan sekitarnya, maka umpan balik (*feed-back*) dapat berasal dari *output*, juga bisa berasal dari lingkungan sistem yang dimaksud.

Dengan demikian defenisi ini akan mempunyai peranan yang penting di dalam pendekatan untuk mempelajari suatu sistem. Pendekatan yang lebih menekankan pada elemen atau komponennya didefenisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu. Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan menentukan dalam mendefenisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Dapat disimpulkan sistem adalah kegiatan-kegiatan yang saling berhubungan antara satu sama lainnya yang terdiri dari objek-objek, unsur-unsur atau komponen-komponen sehingga membentuk suatu kesatuan pemerosesan untuk mencapai tujuan tertentu.

### II.1.1. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Seperti yang telah disebutkan dari beberapa pengertian diatas, maka sistem terdiri dari elemen-elemen yang berkaitan. Dan elemen-elemen terdiri dari :

#### 1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut “supra sistem”.

#### 2. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat pisahkan.

#### 3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut operasi lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan yang menguntungkan

merupakan bagi sistem tersebut. Dengan demikian, lingkungan luar tersebut harus dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

#### 4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lainnya disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian dapat terjadi suatu integrasi sistem untuk membentuk satu kesatuan.

#### 5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh di dalam suatu sistem unit komputer. “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “data” adalah *signal input* untuk diolah menjadi informasi.

#### 6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini

dapat digunakan sebagai masukan untuk pengambil keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain.

#### 7. Pengolah Sistem (*Proses*).

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

#### 8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministik*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. (Tata Sutabri, 2012 : 20-21)

### **II.1.2. Siklus Hidup Pengembangan Sistem**

Siklus hidup sistem (*system life cycle*) merupakan proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem.

Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian, proses ini merupakan aspek yang sangat penting. Kita akan melihat beberapa fase/tahapan dari daur hidup suatu sistem.

1. Mengenal Adanya Kebutuhan.

Sebelum segala sesuatunya terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3. Pemasangan Sistem

Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang akan merupakan langkah akhir dari suatu pembangunan sistem.

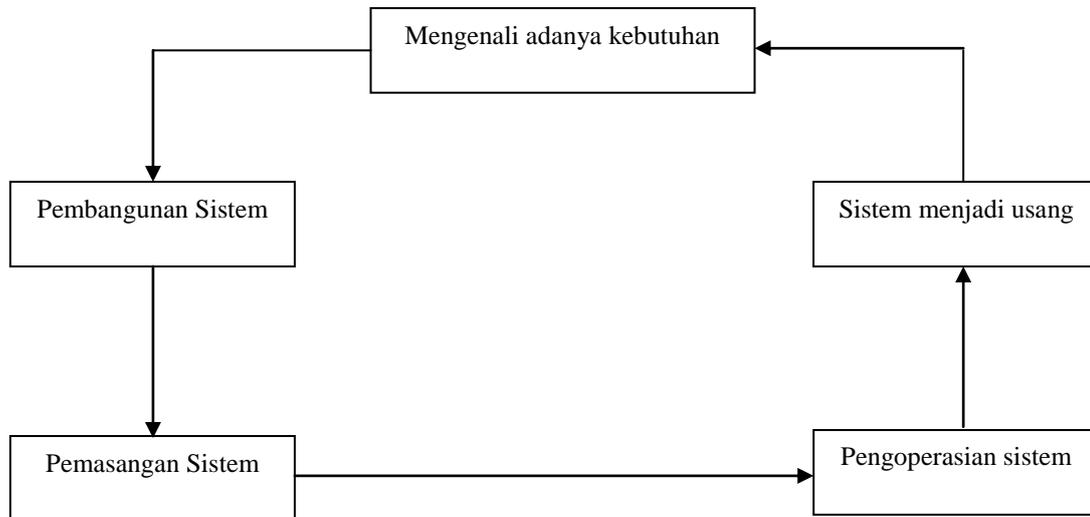
#### 4. Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis. Sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan pengaturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

#### 5. Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis, sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan yang muncul. Sistem beradaptasi terhadap perubahan-perubahan lingkungannya dinamis. Sampailah pada kondisi dimana sistem tersebut tidak dapat lagi beradaptasi dengan perubahan-perubahan yang ada atau secara ekonomis tidak layak lagi untuk dioperasikan. Sistem yang baru kemudian dibangun untuk menggantikannya. Untuk dapat menggambarkan daur hidup sistem ini, dapat dilihat pada gambar II.1. Sebagai berikut (Tata Sutabri, 2012 : 27-29).



**Gambar II.1. Daur Hidup Sistem**

*Sumber : Tata Sutabri (2012 : 29)*

## II.2. Pengertian Informasi

Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. (Tata Sutabri, 2012 : 29)

## II.3. Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri, 2012 : 46).

### II.3.1. Komponen Dan Jenis Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran.

1. Blok Masukan (*Input Block*).

Input mewakili data yang masuk ke dalam sistem informasi. *Input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan "*tool box*" dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan

mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi sistem terdiri dari 3 (tiga) bagian yaitu teknisi teknologi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*database management system*).

6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri, 2012 : 47 - 48 ).

## **II.4. Sistem Informasi Geografis (SIG)**

### **II.4.1. Pengenalan SIG**

Pada dasarnya, istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografis. Dengan demikian, pengertian terhadap tiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG.

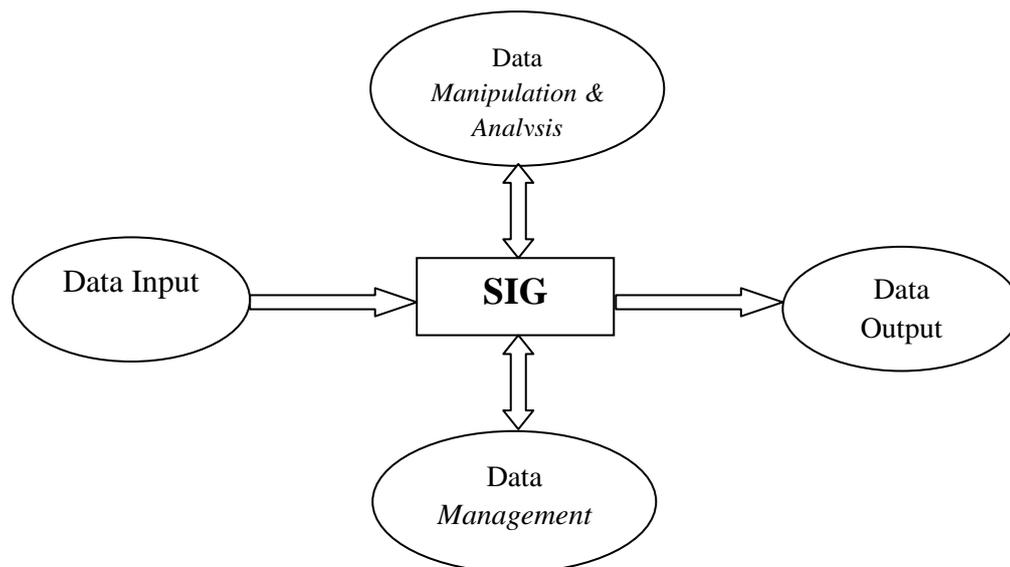
Istilah “Geografis” merupakan bagian dari spasial (keruangan). Penggunaan kata “geografis” mengandung pengertian suatu persoalan atau hal mengenai (wilayah dipermukaan) bumi: baik permukaan dua dimensi atau tiga dimensi. Dengan demikian, istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak dipermukaan bumi, atau informasi mengenai keterangan-keterangan (atribut) objek penting yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui.

Dengan memperhatikan pengertian sistem informasi ,maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenan dengan obojek-objek penting yang terdapat di permukaan bumi. Jadi, SIG juga merupakan sejenis perangkat lunak, perangkat keras (manusia, prosedur, basis data, dan fasilitas jaringan komunikasi) yang dapat digunakan untuk memfasilitasi proses pemasukan, penyimpanan, manipulasi, menampilkan dan keluaran data/informasi geografis berikut atribut-atribut terkait.

#### II.4.2. Subsistem SIG

SIG dapat diuraikan menjadi beberapa sub-sistem sebagai berikut :

1. Data Input : sub-sistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya di berbagai sumber. Sub-sistem ini pula yang bertanggung jawab dalam mengoperasikan atau mentransformasikan format-format dari aslinya ke dalam format (*native*) yang dapat digunakan oleh perangkat SIG yang bersangkutan.
2. Data Output : sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, *report*, peta, dan lain sebagainya.
3. Data Management : sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa sehingga mudah dipanggil kembali atau *retrieve* (di-load ke memori), di-update dan di-*edit*.
4. Data Manipulation & Analysis : sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu sub-sistem ini juga melakukan manipulasi ( evaluasi dan penggunaan fungsi-fungsi dan operator matematis dan logika) dan pemodelan untuk menghasilkan informasi yang diharapkan (Eddy Prahasta (2009 : 118)).



**Gambar II.2. Ilustrasi Sub-Sistem SIG**

*Sumber : Eddy Prahasta (2009 : 119)*

### II.4.3. Komponen-Komponen SIG

SIG merupakan salah satu sistem yang kompleks dan pada umumnya juga terintegrasi dengan lingkungan sistem komputer lainnya ditingkat fungsionalitas dan jaringan (network). Jika diuraikan, SIG sebagai sistem terdiri dari beberapa komponen sebagai berikut.

#### 1. Perangkat Keras

Perangkat keras yang sering digunakan untuk aplikasi SIG adalah *digitizer, printer, plotter, Central Processing Unit (CPU), mouse, printer, dan scanner.*

#### 2. Perangkat Lunak

Perangkat lunak yang digunakan untuk aplikasi SIG cukup banyak diantaranya : ArcView GIS, Quantum GIS, MapServer, ILWIS, MapInfo dan lain-lain.

### 3. Data dan Informasi geografi

Data dan informasi yang diperlukan baik secara tidak langsung dengan cara meng-*import*-nya dari perangkat-perangkat lunak SIG yang lain maupun secara langsung dengan cara melakukan digitasi data spasial dari peta dan memasukkan data atributnya dari tabel-tabel dan laporan dengan menggunakan *keyboard*. (Eddy Prahasta, 2009 : 120)

## II.5. Peta

Pemahaman mengenai Sistem Informasi Geografis (SIG) tidak terlepas dari adanya peta. Sebuah peta (*hardcopy*) selalu menyediakan gambar atau simbol unsur-unsur geografi dengan bentuk, pola, ukuran dan warna yang tetap atau statis (tidak ada berubah) meskipun oleh para penggunanya (akan) diperlukan untuk memenuhi berbagai kebutuhan (yang kemungkinan besar berbeda).

Di dalam konteks yang lebih luas, peta juga merupakan salah satu aset public yang sangat berharga. Survei-survei pemetaan yang telah dilakukan di berbagai negara telah mengindikasikan bahwa nilai keuntungan (*benefit*) atas penggunaan peta akan meningkat jauh bahkan hingga beberapa kali lipat dari biaya produksi peta itu sendiri. Meskipun demikian, jika dibandingkan dengan fungsionalitas peta-peta seperti ini (dalam bentuk analog dan *hardcopy*), SIG memiliki beberapa keunggulan *inherent* karena penyimpanan data (atau informasinya) dan presentasinya dipisahkan secara tegas atau dibedakan dengan jelas. Dengan demikian, data (basis data spasial) yang dimiliki oleh SIG dapat dipresentasikan dalam berbagai cara dan bentuk (dinamis).

## II.6. Kamus Data

Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*), dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Yang berisi :

1. Nama-nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait
3. Deskripsi merupakan deskripsi data.
4. Informasi tambahan- seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data. (Rosa A.S, M. Shalahuddin ; 2011 : 67)

**Tabel II.3 Simbol Untuk Menjelaskan Informasi Tambahan**

Simbol	Keterangan
=	di susun atau terdiri dari
+	Dan
[]	baik...atau...
{ } <sup>n</sup>	n kali di ulang / bernilai banyak
()	data opsional
*...*	batas komentar

*Sumber : Rosa A.S, M. Shalahuddin (2011 : 68)*

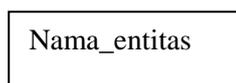
## II.7. Entity Relationship Diagram (ERD)

ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang menghubungkan antara objek yang satu dengan objek yang lainnya dalam dunia nyata. Yang terdiri dari 3 komponen, yaitu : entitas (*entity*), atribut (*attribute*), relasi (*relationship*). ( Ema Utami, anggit Dwi Hartanto , 2012 : 18).

### II.7.1. Komponen-komponen ERD

#### 1. Entitas

Merupakan suatu “objek nyata” yang mapu dibedakan dengan objek lainnya. Entitas digambarkan sebagai bentuk persegi panjang dengan nama entitas terletak di dalamnya. Jika nama entitas terdiri lebih dari satu suku kata, maka diberikaan tanda garis bawah ( \_ ) untuk menyambungunya. ( Ema Utami, anggit Dwi Hartanto , 2012 : 18).



**Gambar II.3. Entitas**

*Sumber : Ema Utami, anggit Dwi Hartanto (2012 : 19)*

#### 2. Atribut

Merupakan semua informasi yang berkaitan dengan entitas. Didalam pemograman, atribut adalah properti dari suatu objek. Atribut di gambarkan dengan lingkaran dengan nama atribut ditulis di tengahnya. ( Ema Utami, anggit Dwi Hartanto , 2012 : 20).

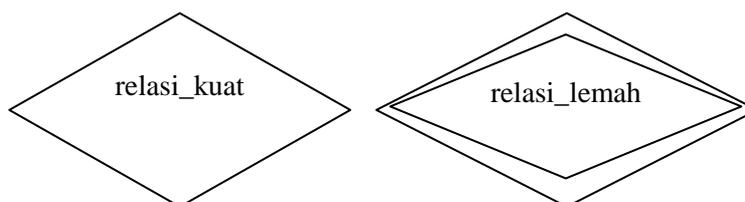


**Gambar II.4. Atribut**

*Sumber : Ema Utami, anggit Dwi Hartanto (2012 : 20)*

### 3. Relasi

Belah ketupat merupakan penggambaran hubungan (relasi) antar entitas atau sering di sebut kerelasian. Ada dua macam penggambaran relasi yakni,kuat dan relasi lemah. Relasi kuat untuk menghubungkan antar entitas kuat, sedangkan relasi lemah untuk menghubungkan antar entitas kuat dengan entitas lemah. ( Ema Utami, anggit Dwi Hartanto , 2012 : 24).



**Gambar II.5. Relasi**

*Sumber : Ema Utami, anggit Dwi Hartanto (2012 : 24)*

## **II.8. Unified Modelling Language (UML)**

### **II.8.1. Pengenalan UML**

*UML* singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error

yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

*UML* telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati, 2011 : 6).

### II.8.2. Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*PackageDiagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.

6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

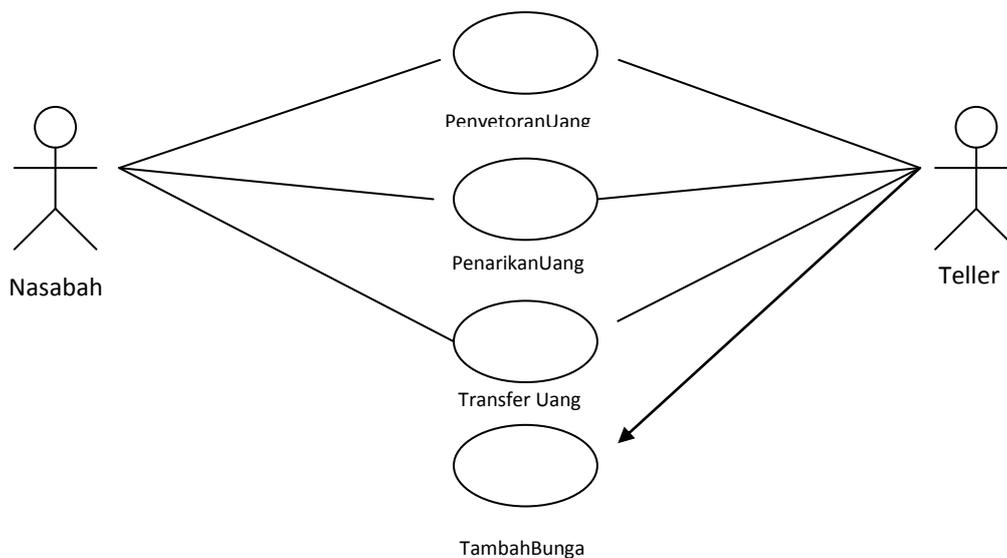
### II.8.3. Diagram Use Case (*use case diagram*)

*Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat.

komponen pembentuk diagram *use case* adalah :

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
3. Hubungan (*Link*), aktor mana saja yang terlibat dalam *use case* ini.

Gambar berikut ini merupakan salah satu contoh bentuk diagram *use case*.

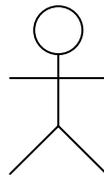


**Gambar II.6. Diagram Use Case**

*Sumber : Probowo Pudjo Widodo (2011:17)*

## 1. Aktor

Sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.



**Gambar II.7. Aktor**

*Sumber : Probowo Pudjo Widodo (2011:17)*

## 2. Use Case

*Use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. *Use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.



**Gambar II.8. Simbol Use Case**

*Sumber : Probowo Pudjo Widodo (2011:22)*

*Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu untuk menghasilkan *use case* yang baik yakni :

- a. Pilihlah nama yang baik

*Use case* adalah sebuah *behaviour* (prilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

- b. Ilustrasikan perilaku dengan lengkap.

*Use case* dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda mengetahui tujuannya. Sebagai contoh memilih tempat tidur (*King Size*, *Queen Size*, atau dobel) saat tamu memesan tidak dapat dijadikan *use case* karena merupakan bagian dari *use case* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

- c. Identifikasi perilaku dengan lengkap.

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

- d. Menyediakan *use case* lawan (*inverse*)

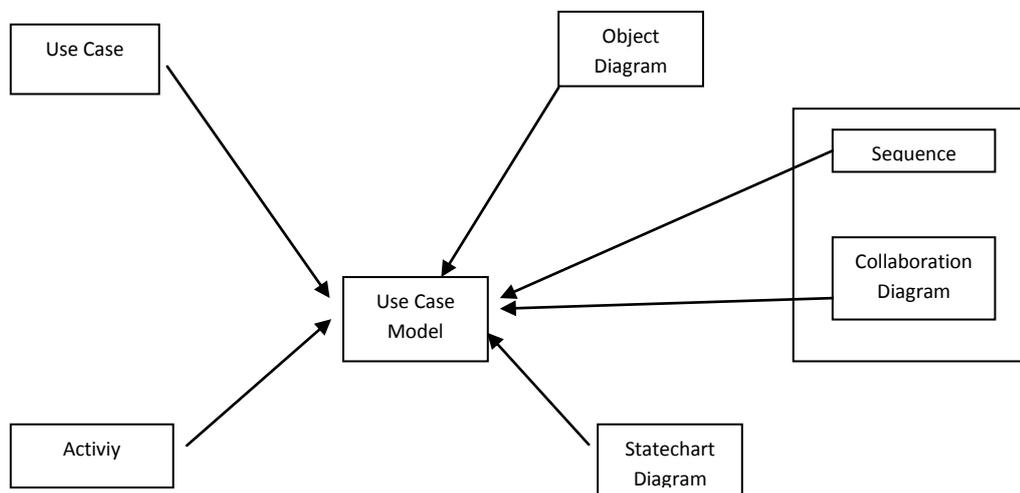
Kita biasanya membutuhkan *use case* yang membatalkan tujuan, misalnya pada *use case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pesanan kamar.

- e. Batasi use case hingga satu perilaku saja.

Kadang kita cenderung membuat *use case* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah use case kita hanya fokus pada satu hal. Misalnya, penggunaan *use case check in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda.

### 3. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Probowo Pudji Widodo; 2011 : 37).



**Gambar II.9. Hubungan Diagram Kelas Dengan Diagram *UML* lainnya**

*Sumber : Probowo Pudjo Widodo (2011 : 38)*

#### 4. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo, 2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

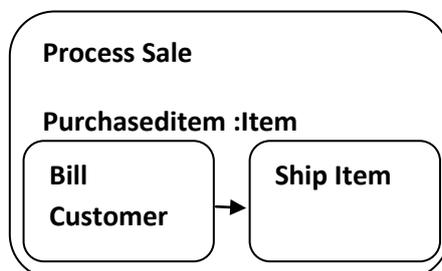
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



**Gambar II.10. Aktivitas sederhana tanpa rincian**

*Sumber : Probowo Pudjo Widodo (2011:145)*

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



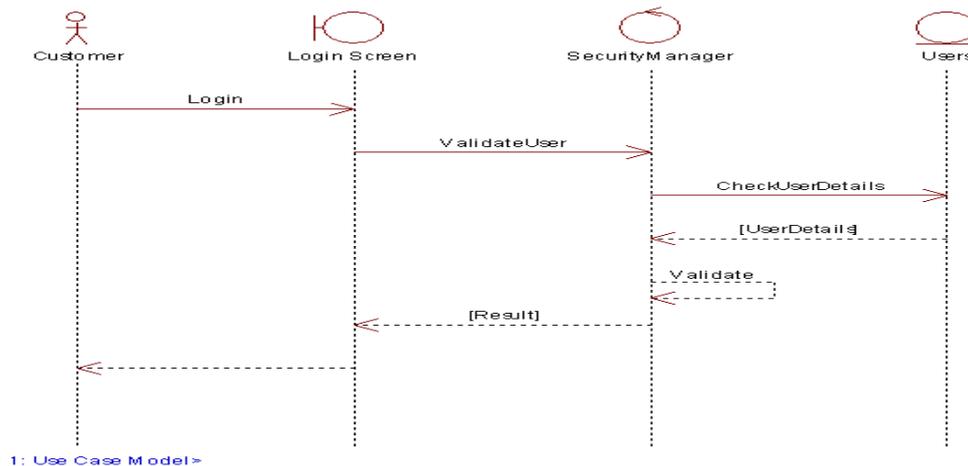
**Gambar II.11. Aktivitas dengan detail rincian**

*Sumber : Probowo Pudjo Widodo (2011:145)*

## 5. *Sequence Diagram*

Ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.12. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya.



**Gambar II.12. Diagram Urutan**

*Sumber : Probowo Pudjo Widodo*

## II.9. PHP

PHP (*PHP Hypertext Preprocessor*) merupakan bahasa bentuk skrip yang ditempatkan dalam server dan di proses di server. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, anda bisa menampilkan isi *database* ke halaman web ( Abdul Kadir, 2008 : 2).

## II.10. MySQL

MySQL adalah salah satu program yang dapat digunakan sebagai database, dan merupakan salah satu software untuk database server yang banyak digunakan. MySQL bisa dijalankan diberbagai platform misalnya windows, linux, dan lain sebagainya. MySQL tergolong sebagai DBMS (*Database Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat.

Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut.

1. Menyimpan data kedalam tabel,
2. Menghapus data dalam tabel,
3. Mengambil data yang tersimpan dalam tabel,
4. Memungkinkan untuk memilih data tertentu yang diambil,
5. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

(Abdul Kadir , 2010 : 10).

MySQL memiliki beberapa kelebihan, antara lain :

1. MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
2. MySQL memiliki kecepatan yang bagus dalam menangani query sederhana.
3. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah Select dan Where dalam perintah query.

4. MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perijinan yang mendetail secara sandi terenkripsi.
5. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta kurang lebih 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
6. MySQL dapat melakukan koneksi dengan klien menggunakan protocol TPC/IP, Unix socket (UNIX), atau Named Pipes (NT).
7. MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa.
8. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
9. MySQL didistribusikan secara open source, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

## **II.11. ArcView**

*ArcView* adalah software yang di keluarkan oleh *ESRI (Environmental Systems Research Institute)*. Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *ArcView* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *ArcView* banyak diterapkan dalam berbagai

pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persil, pengendalian dampak lingkungan, bahkan untuk keperluan militer (Eko Budiyanto : 2010: 176).

*Arcview* di dukung dari skrip Avenue. Melalui avenue ini dapat dibentuk suatu kemampuan baru pada *arcview*. Avenue dapat digunakan untuk merombak wajah *Arcview* sesuai kebutuhan penggunanya.

Avenue adalah sebuah objek (*Object Oriented Programming*). Dengan avenue ini dapat dibentuk sebuah interface baru pada *Arcview*, otomatis pekerjaan-pekerjaan yang bersifat berulang ataupun membuat sebuah alur analisis spasial khusus yang belum terdapat pada *Arcview* tersebut. Avenue banyak digunakan untuk membenyuk sebuah sistem informasi aplikasi pada suatu instansi dengan berbasis *Arcview* GIS.