

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali diuraikan oleh Kenneth boulding terutama menekan pentingnya perhatian terhadap setiap bagian yang membentuk sistem.

Kata sistem berasal dari bahasa Yunani yaitu "*Systema*" yang berarti kesatuan. Sistem adalah sekumpulan komponen yang saling berhubungan yang harus bekerja bersama-sama untuk menghasilkan suatu kesatuan metode, prosedur teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai tujuan.

Sistem adalah sekelompok unsur yang erat berhubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri ; 2005 : 8)

#### **II.1.1. Pengertian Informasi**

Informasi adalah data yang telah diproses dalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau pun keputusan saat mendatang (Tata Sutabri ; 2005 : 16).

Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri ; 2005 : 23).

### **II.1.2. Pengertian Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam 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 strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem Informasi adalah sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi.

Dengan kata lain, SI merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan (budhi Sutedjo ; 2006 : 11 ).

Sistem informasi terdiri dari komponen-komponen dengan istilah berikut:

1. Blok Masukan (*Input Blok*): *Input* Blok ini mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang dimasukkan 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*): Hasil 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*): Merupakan *Tool Box* (kotak alat) dalam sistem informasi karena digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan
5. Blok Basis Data (*Database Block*): Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
6. Blok Kendali (*Control Block*): Banyak hal yang merusak sistem informasi, seperti api, air, debu dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal tersebut dapat dicegah.  
(Tata Sutabri ; 2005 : 42-43).

## **II.2. Sistem Informasi Geografis (SIG)**

### **II.2.1. Pengertian Sistem Informasi Geografis (SIG)**

SIG merupakan suatu sistem yang terdiri dari perangkat keras, perangkat lunak, data, manusia, organisasi dan lembaga yang digunakan untuk

mengumpulkan, menyimpan, menganalisa dan menyebarkan informasi-informasi mengenai daerah-daerah di permukaan bumi. Dengan metodologi SIG dapat memberikan kontribusi dengan cara memanipulasi data dan melakukan analisis yang diperlukan dengan waktu yang singkat dan murah, dan SIG juga berkontribusi untuk menghasilkan bentuk yang berbeda yang dapat disajikan secara grafis (Mohammad Iqbal; 2012; 204-205)

Komponen – komponen pada Sistem Informasi Geografi ada 4, yaitu :

1. *Hardware*

SIG membutuhkan komputer untuk penyimpanan dan pemrosesan data dengan spesifikasi yang lebih tinggi dibandingkan dengan sistem informasi lainnya, karena data yang digunakan baik data vector maupun data transfer membutuhkan ruang yang besar serta membutuhkan memori besar dan prosesor yang cepat untuk proses analisa.

2. *Software*

SIG harus menyediakan fungsi dan *tool* yang mampu melakukan penyimpanan data, analisis dan menampilkan informasi geografis. Elemen yang harus ada dalam sebuah software SIG adalah :

- a. Tool untuk melakukan input dan transformasi data geografis
- b. Sistem Manajemen Basis Data (DBMS)
- c. Tool yang mendukung query geografis, analisis dan visualisasi.
- d. *Graphical User Interface* (GUI) untuk memudahkan akses pada *tool* geografi.

### 3. Data

Data terdiri dari data spasial, dan data atribut. Data spasial adalah data yang terdiri dari lokasi eksplisit suatu geografi yang diset ke dalam bentuk koordinat. Data atribut adalah gambaran data yang terdiri atas informasi yang relevan terhadap suatu lokasi, seperti kedalaman, ketinggian dan lainnya.

### 4. Metode

GIS harus memiliki keserasian antara rencana desain yang baik dan aturan dunia nyata. Metode yang tepat akan memberikan ketepatan model dan implementasi yang tergantung pada permasalahan yang ada.

### 5. Manusia

Manusia melakukan perawatan dan pemanfaatannya secara baik dan benar agar tujuan sistem tercapai (Much Aziz Muslim; 2005; 119-120)

## II.2.2. Subsistem GIS

Sistem informasi geografis dapat diuraikan menjadi beberapa subsistem:

1. *Data Input*. Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini bertanggung jawab dalam mengkonversi atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan SIG.
2. *Data Output*. Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk *softcopy* maupun *hardcopy* seperti *table*, grafik, peta dan lain-lain.

3. *Data Management*. Subsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah basisdata sedemikian rupa sehingga dipanggil, di-*update* dan di-*edit*.
4. Data Manipulasi dan Analisis. Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan (Mohammad Iqbal; 2012; 204-205)

### II.2.3. Konsep Model Data Pada SIG

Terdapat dua model data atau gambar yang digunakan dalam SIG, yaitu :

#### a. Data Vektor

Melakukan proses pengolahan data atau gambar menggunakan garis dan kurva, yang memuat informasi warna, dimensi serta posisi. Vektor bersifat *resolution-independent* atau tidak tergantung pada resolusi. Artinya, vektor dapat diubah-ubah baik bentuk, ukuran, posisi atau warnanya pada resolusi berapapun tanpa mengubah kualitas tampilannya. Vektor dapat pula berupa satu titik tunggal.

#### b. Data Raster

Data ini disebut juga dengan *bitmap*, yaitu gambar yang komposisinya terdiri atas titik-titik berbentuk bujur sangkar, yang dinamakan dengan pixel, yang disusun pada suatu *grid*. Setiap titik-titik pada *grid* tersebut masing-masing mengandung warna tersendiri. Memodifikasi tiap *pixel*. Raster bersifat *resolution dependent* atau bergantung pada resolusi. Artinya data menampilkan gambar yang terpaku pada resolusi tertentu.

Jadi, ketika gambar tersebut diperkecil atau diperbesar, kualitas gambar akan berubah (Mohammad Iqbal; 2012; 204-205)

### **II.3. Peta**

Peta adalah penyajian grafis dari seluruh atau sebagian permukaan bumi pada suatu bidang datar dengan menggunakan suatu skala dan sistem proyeksi tertentu. Secara umum peta terbagi beberapa jenis, yaitu:

#### **a. Peta Umum**

Peta ini adalah peta yang menyajikan gambaran umum yang ada dipermukaan bumi tersebut baik bersifat alamiah (misalnya seperti sungai, danau gunung, pegunungan, laut, hutan, dan lain-lain) maupun bersifat buatan manusia ( misalnya ; batas wilayah, jalan raya, kota, pelabuhan, perkebunan, dan lain-lain ). Contoh peta umum adalah peta dunia, peta negara, dan lain-lain.

#### **b. Peta Khusus**

Peta khusus atau peta tematik merupakan adalah peta yang menyajikan gambaran khusus (spesifik) di permukaan bumi. Peta tematik umumnya digunakan sebagai analisis dari beberapa unsur permukaan bumi didalam pengambilan keputusan. Contoh peta khusus adalah peta iklim dunia, peta persebaran penduduk, peta tata guna lahan, dan lai-lain.

#### **II.4. ArcView**

Arc View merupakan sebuah software pengelola data spasial. Software ini memiliki berbagai keunggulan yang dapat dimanfaatkan oleh kalangan pengolah data spasial. Arc View memiliki kemampuan dalam pengolahan atau editing arc, menerima atau konversi dari data lain seperti CAD, atau dihubungkan dengan data image seperti format JPG, TIFF, atau image gerak.

Untuk memulai penggunaan software Arc View, panggil program ini dari start menu.

- Klik Start
- Pilih program
- Pilih ESRI
- Pilih Arc View GIS

Cara lain adalah dengan klik ganda pada shortcut Arc View di desktop. Selanjutnya Arc View akan menanyakan suatu proyek baru atau memanggil proyek yang sudah ada. Pembuatan proyek baru dilakukan dengan memilih opsi With a new View jika telah terdapat proyek yang akan diolah lebih lanjut pilih open an Existing Project. Hasil pengolahan data spasial dalam Arc View disimpan dalam sebuah proyek dengan ekstensi APR (Eko Budiyanto ; 2005 : 9).

#### **II.5. *Unified Modeling Language* (UML)**

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘beorientasi objek pemodelan (*modeling*)’ sesungguhnya digunakan untuk penyederhanaan permasalahan-

permasalahan yang kompleks sedemikian rupa sehingga mudah dipahami dan dipelajari.

UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch, Object Modeling Technique* (Munawar ; 2005 : 17) .

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan suatu aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Adapun *view* dan diagram UML yang ada seperti pada tabel II.1.

**Tabel II.1 View dan Diagram UML**

<i>View</i>	<i>Diagram</i>	<i>Main concepts</i>
<i>Static view</i>	<i>Class diagram</i>	<i>Class, association, generalization, dependency, Relization, interface</i>
<i>Use case view</i>	<i>Use case diagram</i>	<i>use case, actor, association, extend, include, use case generalization</i>
<i>Implementation view</i>	<i>Component diagram</i>	<i>Component, interface, dependency, Realization</i>
<i>Deployment view</i>	<i>Deployment diagram</i>	<i>Node, component, dependency, location</i>
<i>State machine</i>	<i>Statechart</i>	<i>State, event, transition, action</i>

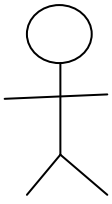
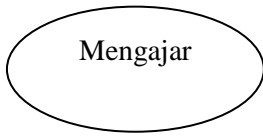
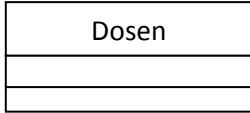
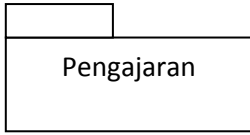
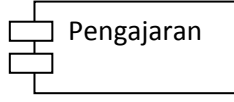

<i>view</i>	<i>diagram</i>	
<i>Activity view</i>	<i>activity diagram</i>	<i>State,activity,completion,transition, fork,join</i>
<i>Interaction view</i>	<i>Sequence diagram</i>	<i>Interaction,object,messages,activation</i>
	<i>Collaboration diagram</i>	<i>Collaboration,interaction,collaboration role,message</i>
Model management view	Class diagram	Package,subsystem,model
All	All	Constraint,stereotype,tagged values

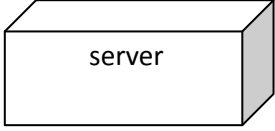
**Sumber: Munawar ; 2005 : 23**

### II.5.1 Pengklasifikasi (*Classifier*)

Pengklasifikasi (*classifier*) pada prinsipnya merupakan konsep diskret dalam model yang memiliki identitas (*identity*), *state*, perilaku (*behaviour*), serta relasi yang dengan pengklasifikasi yang lain (*relationship*). Adapun beberapa bentuk pengklasifikasi (*classifier*) adalah seperti gambar II.2.

Tabel II.2. Beberapa pengklasifikasi (*Classifier*).

Pengklasifikasi	Kegunaan	Notasi
<i>Actor</i>	Menggambarkan semua objek diluar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan	
<i>Use case</i>	Menggambarkan fungsionalitas yang dimiliki sistem	
<b>Kelas (<i>class</i>)</b>	Menggambarkan konsep dasar pemodelan sistem	
<b>Subsistem (<i>subsystem</i>)</b>	Menggambarkan paket spesifikasi serta implementasi	
<b>Komponen (<i>component</i>)</b>	Menggambarkan bagian-bagian fisik sistem/perangkat lunak yang dikembangkan	
<b>Antar muka (<i>interface</i>)</b>	Menggambarkan antar muka pengiriman pesan ( <i>mesagge</i> ) antar pengklasifikasi	 Antar muka pengajar- mahasiswa

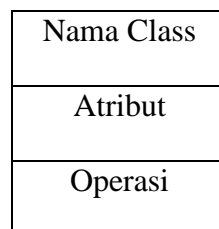
<b>Simpul(<i>node</i>)</b>	Menggambarkan sumber daya komputasional yang digunakan oleh sistem.	
----------------------------	---	---

**Sumber: (Adi nugroho ; 2010 : 16)**

### II.5.1.1 Jenis-jenis UML

#### 1. *Class*

Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu class beserta dengan atribut dan operasinya. Class adalah pembentuk utama dari sistem yang berorientasi objek. Adapun contoh notasi class seperti pada gambar II.1 adalah sebagai berikut :



**Gambar II.1. Notasi Class pada UML**

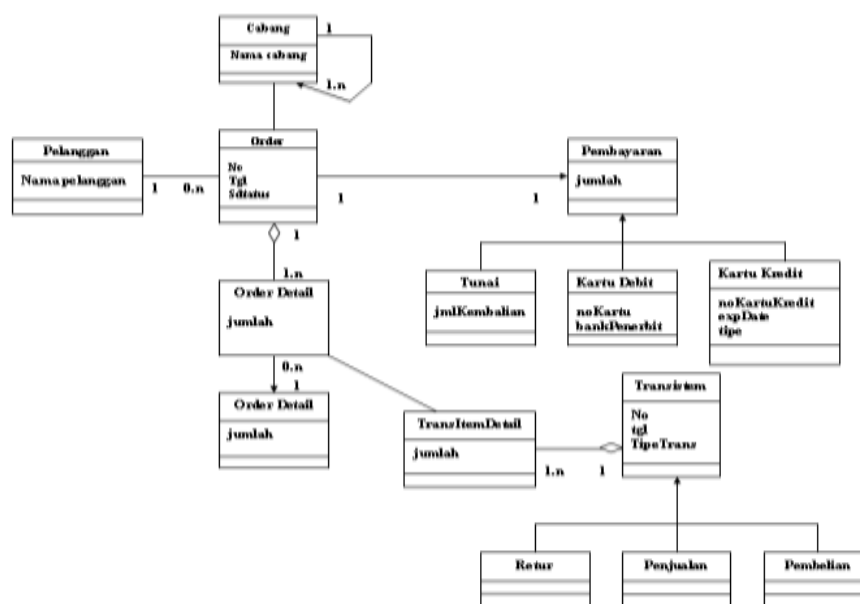
**Sumber : ” Pemodelan Visual dengan UML (Munawar ; 2009 : 41)”**

#### 2. *Class Diagram*

Sama seperti *class*, maka class diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan

antar *class* yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

*Class* diagram umumnya tersusun dari elemen *class*, *interface*, *dependency*, *Generalization* dan *Association*. Relasi *dependency* menunjukkan bagaimana terjadi ketergantungan antar *class* yang ada. Relasi *Generalization* menunjukkan bagaimana suatu *class* menjadi *superclass* dari *class* lainnya dan *class* tersebut menjadi *subclass* dari *class* tersebut. Relasi *Association* menggambarkan navigasi antar *class*, berapa banyak objek lain bisa berhubungan dengan satu objek (*multiplicity* antar *class*), dan apakah satu *class* menjadi bagian dari *class* lainnya (*agregation*). *Class diagram* digunakan untuk menggambarkan disain statis dari sistem yang sedang dibangun. Adapun Contoh *Class diagram* seperti pada gambar II.2 adalah sebagai berikut:

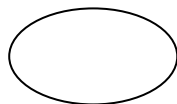


**Gambar II.2. Model Class Diagram**

Sumber : ” (Munawar ; 2005 : 220)”

### 3. *Use Case*

*Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah – langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Adapun contoh notasi *use case* dapat dilihat pada gambar II.3 adalah sebagai berikut :

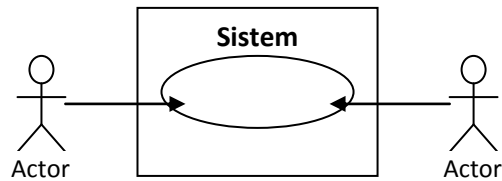


**Gambar II.3. Notasi *Use Case* pada UML**

**Sumber : "Pemodelan Visual dengan UML (Munawar ; 2009 : 42)"**

### 4. *Use Case Diagram*

*Use Case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* mempresentasikan sebuah interaksi antara actor dengan sistem. *Use Case* menggambarkan kata kerja seperti *Login* ke sistem, *maintenance user* dan sebagainya. Adapun contoh model *use case* diagram seperti pada gambar II.4 adalah sebagai berikut :



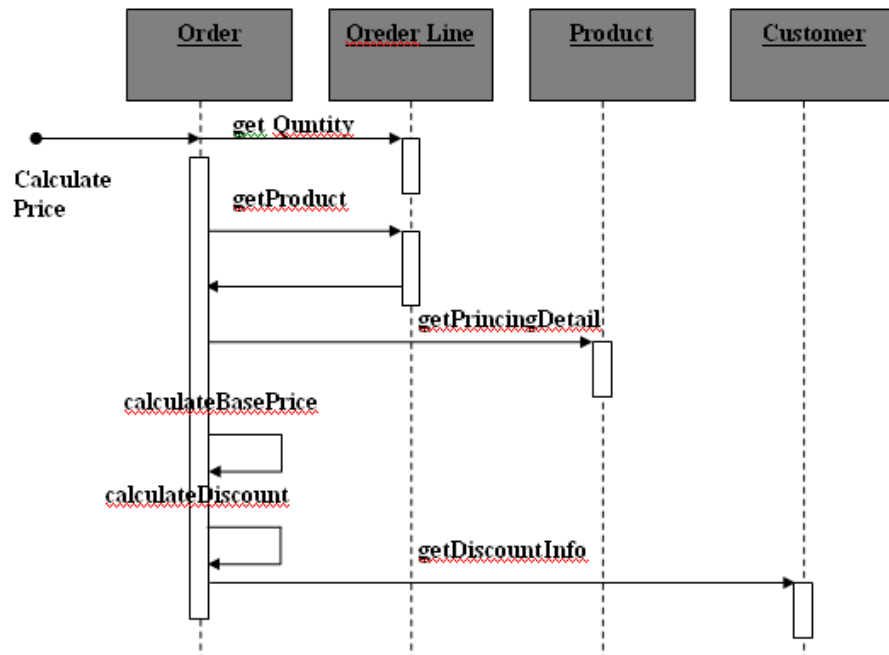
**Gambar II.4. Model Use Case pada UML**

**Sumber : ” Pemodelan visual dengan UML (Munawar ; 2009 : 43)”**

## 5. *Sequence Diagram*

*Sequence diagram* menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan *sequence diagram* merupakan aktivitas yang paling kritical dari proses disain karena artefak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program.

*Sequence diagram* biasanya tersusun dari elemen objek, *Interaction* dan *Message*. *Interaction* menghubungkan 2 Obyek dengan pesannya. Diagram ini menjelaskan aspek dinamis dari sistem yang sedang dibangun. Di dalam *sequence diagram*, terdapat kelas *boundary*, *control* dan *entity*. Adapun Contoh *sequence diagram* seperti pada gambar II.5 adalah sebagai berikut:



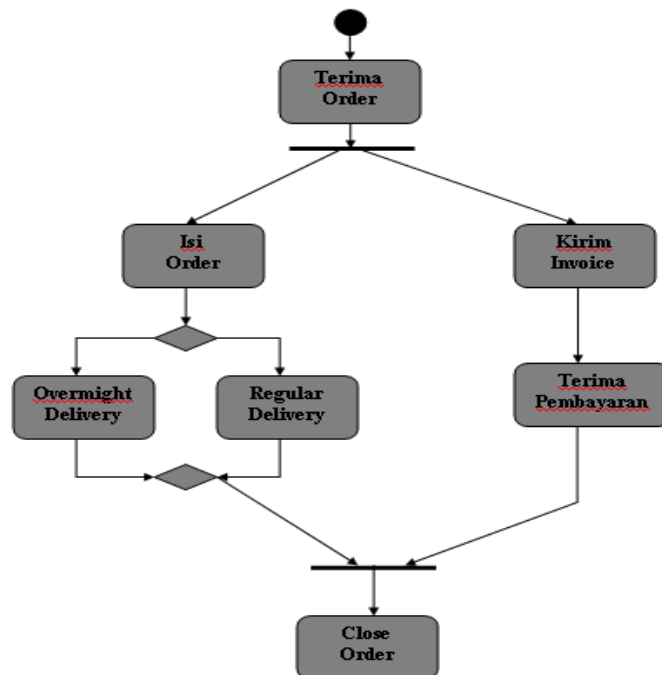
**Gambar II.5. Model Sequence Diagram**

**Sumber : ” Pemodelan visual dengan UML (Munawar ; 2005 : 91)”**

## 6. Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standart UML

menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu, digambarkan dengan simbol belah ketupat. Untuk mengilustrasikan proses-proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Adapun Contoh *activity diagram* seperti pada gambar II.6 adalah sebagai berikut:



**Gambar II.6. Model Activity Diagram**

Sumber : ” (Munawar ; 2005 : 111)”

### II.5.2. Pengertian Database

Berikut ini pengertian data base yang diberikan oleh James Martin dalam bukunya “*Database Organization*” sebagai berikut :

*Database* adalah suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu dengan yang lain atau tidak perlu suatu kerangkapan data (*controlled redundancy*) dengan cara tertentu sehingga mudah digunakan atau ditampilkan kembali; dapat digunakan oleh satu atau lebih program aplikasi secara optimal; Data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya; Data disimpan sedemikian rupa sehingga penambahan, pengambilan, dan modifikasi dapat dilakukan dengan mudah dan terkontrol.

Dari pengertian tersebut dapat disimpulkan bahwa sistem database mempunyai beberapa kriteria yang penting yaitu:

- A. Bersifat data oriented dan bukan program oriented.
- B. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah databasenya.
- C. Dapat berkembang dengan mudah baik volume maupun strukturnya.
- D. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
- E. Dapat digunakan dengan cara yang berbeda.
- F. Kerangkapan data (*data redundancy*) minimal.

Keenam kriteria tersebut membedakan secara nyata/jelas antara file database dan file tradisional yang bersifat oriented, yaitu bahwa dapat digunakan oleh satu program aplikasi; berhubungan dengan suatu persoalan tertentu untuk sistem yang

direncanakan; perkembangan data hanya mungkin terjadi pada volume data saja; munculkan data terlalu /tidak terkontrol dan hanya dapat digunakan dengan satu cara tertentu saja. Selanjutnya James F. Courtney Jr. Dan David B. Paradise dalam buku “ Database System for Management “ menjelaskan :

Sistem database adalah sekumpulan database yang dapat dipakai secara bersama-sama, personal-personal yang merancang dan mengolah database, teknik-teknik

Dua tujuan utama dari konsep database adalah meminimumkan pengulangan dan mencapai independensi data. Independensi adalah kemampuan untuk membuat perubahan dalam struktur data tanpa membuat perubahan pada program yang memproses data. Independensi data dicapai dengan menempatkan spesifikasi dalam tabel dan kaus terpisah secara fisik dan program. Program mengacu pada tabel untuk mengakses data. Perubahan struktur data hanya dilakukan sekali, yaitu dalam tabel. Suatu perusahaan mengadopsi konsep database dan hirarki data menjadi:

- Database
  - File
    - Catatan
      - Elemen Data

File-file sendiri dapat tetap ada, mewakili komponen-komponen utama dari database; namun organisasi fisik dari data tidak menghambat pemakaian. Tersedia berbagai cara untuk mengintegrasikan isi file-file yang memiliki hubungan logis.

(Tata sutabri ; 2005 : 161-162).

## II.6. Hyper Text Transfer Protocol (HTTP)

Web merupakan terobosan baru sebagai teknologi informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di internet. Pengguna hanya perlu mengklikkan tombol *mouse* pada *link-link hyper text* yang ada untuk mengakses kedokumen-dokumen di berbagai lokasi di internet. *Link-linknya* sendiri dapat mengacu kepada dokumen web, server FTP (*File Transfer protocol*), *e-mail*, ataupun layanan lainnya.

Server dan *browser* web berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini, yaitu HTTP. HTTP bertugas menangani perintah-perintah (*Request*) dari browser untuk mengambil dokumen-dokumen web.

HTTP dapat dianggap sebagai sistem yang bermodel *Client-Server*. *Browser web*, sebagai *Client*, mengirimkan permintaan kepada server web untuk mengirimkan dokumen-dokumen web yang dikehendaki pengguna. Server web lalu memenuhi permintaan ini dan mengirimkannya melalui jaringan kepada *browser*. Setiap permintaan akan dilayani dan ditangani sebagai suatu koneksi terpisah yang berbeda.

## II.7. Hyper Text Markup Language (HTML)

Dewasa ini dikenal dengan bahasa standar untuk membuat dokumen web. Sesungguhnya *Hypertext Markup Language* (HTML) justru tidak dibuat untuk mempublikasikan informasi di web, namun oleh karena kesederhanaan serta

kemudahan penggunaannya, HTML kemudian dipilih untuk mendistribusikan informasi di web.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*, masing-masing ditandai oleh pasangan *container tag* `<head>` dan `<body>`. Bagian *head* berisikan judul dokumen dan informasi-informasi dasar lainnya, sedangkan bagian *body* adalah data dokumennya. Pengaturan format teks dan pembentukan *link* dilakukan terhadap obyeknya langsung ditandai dengan oleh *tag-tag* HTML.

## II.8. MapServer

MapServer merupakan suatu aplikasi server yang memungkinkan suatu halaman web dapat memuat suatu peta yang mengandung informasi seperti Sistem Informasi Geografis. MapServer dibangun berdasarkan atas kebutuhan untuk membangun aplikasi SIG diatas yang berbasis web. Dengan menggunakan MapServer diharapkan suatu peta dapat memberikan informasi secara lengkap dan tepat, karena sistem yang berbasis web akan lebih mudah dan cepat diakses oleh pengguna dari berbagai tempat hanya dengan menggunakan komputer, web-browser dan jaringan internet.

MapServer adalah salah satu program aplikasi CGI (*Common Gateway Interface*) yang dapat digunakan untuk membangun aplikasi Sistem Informasi Geografis berbasis web (*WebGIS*). MapServer menyediakan beberapa fitur yang diperlukan untuk pengolahan suatu peta, misalnya memperbesar atau memperkecil ukuran skala peta, pengolahan informasi yang berkaitan dengan

suatu lokasi dan lainnya. MapServer membantu penyajian suatu peta yang lebih interaktif, dimana pengguna dapat mengakses informasi geografis yang lengkap hanya dengan menggunakan komputer, *web-browser* dan jaringan internet.

Dalam penyusunan skripsi ini, penulis merancanr suatu perangkat lunak yang akan dapat memberikan informasi peta jalur trayek angkutan umum kota Medan. Aplikasi yang dirancang akan menggunakan teknologi SIG yaitu MapServer khususnya paket MS4W yang bekerja pada sistem operasi Windows.

MapServer merupakan aplikasi *freeware* dan *open source* yang memungkinkan pemakai menampilkan data spasial (peta) di web. Aplikasi ini pertama kali dikembangkan di Universitas Minnessota, Amerika Serikat untuk proyek ForNet (sebuah proyek untuk manajemen sumber daya alam) yang disponsori NASA (*National Aeronautics and Space Administration*). Dukungan NASA dilanjutkan dengan dikembangkannya proyek TerraSIP untuk manajemen data lahan. Saat ini, karena sifatnya yang terbuka (*open source*), pengembangan MapServer dilakukan oleh pengembang diberbagai Negara.

Pada bentuk paling dasar MapServer berupa sebuah program CGI (*Common Gateway Interface*). Program tersebut akan dieksekusi di web server dan beberapa parameter tertentu (terutama konfigurasi dalam bentuk file \*.MAP) akan menghasilkan data yang kemudian akan dikirim ke web *browser*, baik dalam bentuk gambar peta ataupun bentuk lain.

### II.8.1. Arsitektur MapServer

Interaksi antara klien dengan server berdasar scenario *request* dan respon. Web *browser* di sisi klien mengirim *request* ke server web. Karena server web tidak memiliki kemampuan pemrosesab peta, maka *request* berkaitan dengan pemrosesan peta akan diteruskan oleh server web ke server aplikasi dan MapServer. Hasil pemrosesan akan dikembalikan lagi melalui server web, terbungkus dalam bentuk file HTML atau applet.

Arsitektur aplikasi pemetaan di web dibagi menjadi dua pendekatan sebagai berikut :

a. Pendekatan *Thin Client*

Pendekatan ini memfokuskan diri pada sisi server. Hampir semua proses dan analisis data dilakukan berdasarkan permintaan (*request*) di sisi server. Data hasil pemrosesan kemudian dikirimkan ke klien dalam format standar HTML, yang di dalamnya terdapat file gambar dalam format standar (misalnya GIF, PNG atau JPG) sehingga dapat dilihat menggunakan sembarang web *browser*. Kelemahan utama pendekatan ini menyangkut keterbatasan pilihan interaksi dengan pengguna yang kurang fleksibel.

b. Pendekatan *Thick Client*

Pada pendekatan ini, pemrosesan data dilakukan di sisi klien menggunakan beberapa teknologi seperti control ActiveX atau applet. Kontrol ActiveX atau applet akan dijalankan di klien untuk

memungkinkan *web browser* dengan kemampuan standar. Dengan adanya pemrosesan di klien, maka transfer data antara klien dengan web server akan berkurang.

MapServer menggunakan pendekatan *thin client*. Semua pemrosesan dilakukan di sisi server. Informasi peta dikirimkan ke *web browser* di sisi klien dalam bentuk file gambar (JPG, PNG, GIF atau TIFF). Untungnya, saat ini kelemahan pendekatan *thin client* dalam hal interaksi dengan pengguna sudah jauh berkurang dengan adanya *framework* aplikasi seperti *Chameleon* atau *CartoWeb*.

## II.8.2. Fitur-fitur Pada MapServer

MapServer mempunyai fitur-fitur sebagai berikut :

- a. Menampilkan data spasial dalam format vector seperti : *Shapefile* (ESRI), ArcSDE (ESRI), PostGIS dan berbagai format vector lain menggunakan Library OGR.
- b. Menampilkan data spasial dalam format raster seperti TIFF/GeoTIFF, EPPL7 dan berbagai format data raster lainnya dengan menggunakan library GDAL.
- c. Dapat dikembangkan (*customizable*), dengan tampilan keluaran yang dapat diatur menggunakan file-file template.
- d. Dapat melakukan seleksi objek berdasarkan nilai, berdasarkan titik area, atau berdasarkan sebuah objek spasial tertentu.
- e. Mendukung *rendering* karakter berupa *font true type*.

- f. Mendukung penggunaan data raster maupun data vector yang di-*tiled* (dibagi-bagi menjadi sub bagian yang lebih kecil sehingga proses untuk mengambil dan menampilkan gambar dapat dipercepat).
- g. Dapat menggambarkan elemen peta secara otomatis : skala grafis, peta indeks dan legenda peta.
- h. Dapat menggambarkan peta tematik yang dibangun menggunakan ekspresi logika atau eksperimen regular.
- i. Dapat menampilkan label dari objek spasial, dengan label dapat diatur sedemikian rupa sehingga tidak saling tumpang tindih.
- j. Konfigurasi dapat diatur secara *on the fly* (dapat disetting dalam keadaan Online) melalui parameter yang ditentukan pada URL.
- k. Dapat menangani beragam system proyeksi secara *on the fly*.

Saat ini, selain dapat mengakses MapServer sebagai program CGI, MapServer juga dapat diakses sebagai modul MapScript, melalui bahasa *script* : PHP, Perl, Python, atau java. Akses fungsi-fungsi MapServer melalui *script* akan lebih memudahkan pengembangan aplikasi.

### **II.8.3. Pengetahuan dasar MapServer**

Dalam pengembangan aplikasi berbasis MapServer, diperlukan beberapa pengetahuan dasar sebagai berikut :

- a. Pengetahuan tentang peta digital antara lain meliputi skala format, bentuk, koordinat dan system proyeksi. Pengetahuan dalam bidang ini sangat

penting karena peta digital memang merupakan data utama yang dikelola oleh MapServer.

- b. Pengetahuan tentang system operasi dan server web pada tempat dimana MapServer akan dipasang.
- c. Pengetahuan tentang struktur dan cara penanganan file berformat HTML. Informasi yang dihasilkan MapServer akan dikirim ke klien (*Web Browser*) dalam format HTML.
- d. Pengetahuan tentang pemrograman di web. Misal dengan menggunakan menggunakan PHP di sisi server atau Javascript di sisi klien. Aplikasi MapServer umumnya bersifat dinamis dan interaktif, sehingga hampir dipastikan pengguna perlu melakukan penyesuaian.
- e. Pengetahuan tentang basis data, karena data spasial hampir tidak pernah lepas dari informasi lain dalam basis data.

#### **II.8.4. Komponen Pembentuk MapServer**

Perkembangan MapServer sebagai sebuah aplikasi *open source*, banyak memanfaatkan aplikasi lain yang juga bersifat *open source*, sedapat mungkin menggunakan aplikasi yang sudah tersedia jika memang memenuhi kebutuhan, untuk menghemat sumber daya dan waktu pengembangan. Pembahasan komponen MapServer terdiri dari empat komponen yaitu :

- a. Komponen untuk akses data spasial.

Komponen ini bertugas untuk menangani baca/tulis data spasial, baik yang tersimpan sebagai file maupun yang tersimpan pada DBMS (*Database Management System*).

- b. Komponen untuk penggambaran peta.

MapServer akan mengirimkan tampilan peta berupa gambar. Pemakai dapat memilih apa format data gambar yang akan digunakan, dan dari beberapa komponen tersebut dapat membentuk gambar peta yang dihasilkan oleh MapServer.

- c. Komponen untuk menangani proyeksi peta.

Digunakan MapServer untuk keperluan menangani system proyeksi peta.

- d. Komponen pendukung.

Misalnya software *editor* seperti *macromedia dreamweaver 8*, dll.

## II.9. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu bersifat *Open Source* pada pelbagai platform.

MySQL termasuk jenis RDBMS (*Relational Database Management System*), seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom (Abdul Kadir ; 2008 : 348).