

BAB II

TINJAUAN PUSTAKA

II.1. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen yang lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandang, seperti contoh sistem yang bersifat abstrak, sistem alamiah, sistem yang bersifat *deterministik* dan sistem yang bersifat terbuka dan tertutup (Tata Sutabri ; 2012 : 22).

II.2. Konsep Dasar Informasi

Belum ada metode untuk mengukur informasi dalam sebuah sistem dan kerumitan informasi tidak memungkinkan adanya suatu rumus atau *algoritma* untuk menghitung isinya. Informasi adalah sebuah istilah yang tidak tepat dalam pemakaiannya secara umum. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi, dan lain sebagainya. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil, dan akhirnya mati. Sistem informasi manajemen berhubungan dengan informasi. Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak

berguna menjadi bentuk berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan (Tata Sutabri ; 2012 : 29).

II.3. Pengolahan Data

Data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai. Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan pengolahan data. Untuk lebih jelasnya akan diuraikan dibawah ini.

1. Penyimpanan data (*data storage*) meliputi pekerjaan pengumpulan (*filing*), pencarian (*searching*), dan memelihara (*maintenance*). Data disimpan dalam suatu tempat yang lazim dinamakan "*file*".
2. Penanganan data (*data handling*) meliputi berbagai kegiatan, seperti pemeriksaan (*verifying*), perbandingan (*comparing*), pemilihan (*sorting*), peringkasan (*extracting*), dan penggunaan (*manipulating*).

Pendekatan sistem berusaha menjelaskan sesuatu yang dipandang dari berbagai sudut pandang serta berusaha menemukan struktur sistem dan proses sistem, seseorang akan dapat menjelaskan mengapa tujuan suatu sistem tidak tercapai. Contohnya, seorang ahli THT yang memahami dengan baik struktur sistem pernafasan dan proses sistem tersebut. Dokter tersebut akan dengan mudah mengidentifikasi penyakit pasiennya yang mengalami kesulitan bernafas. Dia akan dapat dengan mudah disebabkan oleh kesalahan struktur sistem pernafasannya. Dengan melakukan identifikasi yang tepat letak permasalahannya, maka dokter tersebut akan dengan mudah menentukan terapinya. Seorang ahli

pada dasarnya selalu mendekati masalah yang dijumpai dalam bidangnya berdasarkan pendekatan sistem (Tata Sutabri ; 2012 : 6).

II.4. Sistem Informasi Geografis

Sistem informasi geografis mempunyai kemampuan untuk menggabungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisa, dan akhirnya memetakan hasilnya. Data yang diolah pada SIG adalah data spasial yaitu sebuah data yang berorientasi geografis dan merupakan lokasi yang memiliki sistem koordinat tertentu, sebagai dasar referensinya. Sehingga aplikasi SIG dapat menjawab beberapa pertanyaan seperti lokasi, kondisi, tren, pola, dan pemodelan. (Doktafia, Sistem Informasi Geografis [AK 011225] : 2)

II.4.1. Subsistem Sistem Informasi Geografis

Dari definisi tersebut dapat diambil kesimpulan bahwa SIG terdiri atas beberapa subsistem. Subsistem tersebut adalah sebagai berikut :

1. *Data Input*

Subsistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya dari berbagai sumber. Subsistem ini pula yang bertanggung jawab dalam mengonversikan atau mentransformasikan format – format data aslinya ke dalam format yang dapat digunakan oleh perangkat SIG yang bersangkutan.

2. *Data Output*

Subsistem 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, grafik, report, peta dan lain sebagainya.

3. *Data Management*

Subsistem ini mengorganisasikan baik data spasial maupun tabel – tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa sehingga mudah dipanggil kembali atau di-*retrieve*, di-*update* dan di-*edit*

4. *Data Manipulation dan Analysis*

Subsistem ini menentukan informasi – informasi yang dapat dihasilkan oleh SIG. Selain itu subsistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsi – fungsi dan operator matematis dan logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan. (Doktafia, Sistem Informasi Geografis [AK 011225] : 2)

II.4.2. Komponen Sistem Informasi Geografis

Untuk mengoperasikan SIG membutuhkan komponen – komponen berupa perangkat keras (*Hardware*), perangkat lunak (*Software*), Data dan Informasi Geografis.

1. *HardWare*

Ketika GIS (*Geographics Information System*) dibuat berskala besar diperlukan spesifikasi komputer yang besar pula serta *host* untuk *client machine* yang mendukung penggunaan *multiple user*. Hal tersebut disebabkan data yang digunakan dalam GIS baik data vektor maupun data raster penyimpanannya membutuhkan ruang yang besar dan dalam proses analisisnya membutuhkan memori yang besar dan processor yang cepat. Adapun *hardware* yang dibutuhkan yaitu, alat masukan data (*digitizer, scanner, keyboard computer, CD reader, diskette reader*), alat penyimpan dan pengolahan data (komputer dengan *hard disk, tapes or catridge unit, CD writer*).

2. *Software*

Dalam pembuatan GIS diperlukan *software* yang menyediakan fungsi tool yang mampu melakukan penyimpanan data, analisis dan menampilkan informasi geografis. Dengan demikian, elemen yang harus terdapat dalam komponen software GIS adalah tool untuk melakukan input dan transformasi data geografis, sistem manajemen basis data (DBMS), tool yang mendukung query geografis, analisa

dan visualisasi, *Graphical User Interface* (GUI) untuk memudahkan akses pada *tools* geografi.

3. Data dan Informasi geografis

Sistem Informasi geografis (SIG) merupakan perangkat pengelolaan basis data (*Data Base Management System*) dimana interaksi dengan pemakai dilakukan dengan suatu sistem antar muka dan sistem *query* dan basis data dibangun untuk aplikasi *multi user*.

(Rahmat Husein ; 2008 : 5).

II.4.3. Model Data dalam SIG

Data *digital* geografis diorganisir menjadi dua bagian, yaitu Data Spasial dan Data Atribut. Definisi dari kedua bagian tersebut adalah sebagai berikut :

1. Data Spasial

Data spasial diperoleh dari peta *hard copy*, foto udara citra satelit, peta digital dan lainnya. Data spasial disini adalah data berupa gambar yang berhubungan dengan lokasi atau posisi, bentuk dan hubungan antar unsurnya. Pemasukan data vektor dilakukan dengan pendigitasian, sedangkan data raster dilakukan dengan scanning.

Bentuk data spasial, terdiri dari :

- a. Titik, dengan format: sepanjang koordinat (x,y) yang tidak mempunyai dimensi panjang dan luas.
- b. Garis, dengan format: kumpulan pasangan koordinat yang mempunyai titik awal dan titik akhir, serta mempunyai dimensi panjang tetapi tidak mempunyai dimensi luas.

- c. Poligon/Area, dengan format : kumpulan pasanganpasangan koordinat yang mempunyai titik awal dan titik akhir, dimana titik awal dan titik akhir berhimpit atau sama serta mempunyai dimensi panjang dan luas.

2. Data Atribut

Data atribut adalah suatu informasi dari suatu informasi dari suatu data grafis (titik, garis, ataupun area) yang disimpan dalam format data tabular.

Data atribut terdiri dari :

- a. Formulir dan daftar, dengan format : kode *alfabetic*, kode *alfanumeric* dan angka.
- b. Laporan lengkap, dengan format : Kata, kalimat dan keterangan lain.
- c. Keterangan gambar (*grafic chart*), dengan format : kata, angka, keterangan penunjuk, liputan area, keterangan simbol.

(Runi Asmaranto, Sistem Informasi Geografis (SIG) dalam sumber daya air : 16).

II.5. PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman *web* bersifat *serverside*, artinya bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (*Web Server*) sedang hasilnya yang dikirimkan ke komputer *client* (*Web Browser*) dalam bentuk *script* HTML (*Hypertext Mark up Language*).

Konsep kerja PHP diawali dengan satu permintaan suatu halaman web oleh *browser*. Berdasarkan *URL (Uniform Resource Locator)* atau dikenal dengan alamat *internet*, *browser* mendapat alamat dari *webserver*, mengidentifikasi alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mengirimkan isinya ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *web server*, selanjutnya *web server* menyampaikan ke client. Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai data base yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari database, merupakan hal yang mudah untuk diimplementasikan. Sampai saat ini telah banyak database yang telah didukung oleh PHP dan kemungkinan akan terus bertambah. Database tersebut adalah *dBase*, *DBM*, *FilePro*, *mSQL*, *MySQL*, *ODBC*, *Oracle*, *Postgres*, *Sybase*, *Velocis*.

Web server adalah *software* yang menjadi tulang belakang dari *world wide web (www)*. *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*).

Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Contohnya, bila data yang dikirim berupa gambar, *browser* yang hanya mampu menampilkan teks (misalnya *lynx*)

tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja. *Web server*, untuk berkomunikasi dengan *client* – nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*Hypertext Transfer Protocol*). Dengan protokol ini, komunikasi antar *web server* dengan *client* – nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*Hypertext Markup Language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Kata *HyperText* mempunyai arti bahwa seorang pengguna *internet* dengan *web browser* – nya dapat membuka dan membaca dokumen – dokumen yang ada dalam komputernya atau bahkan jauh tempatnya sekalipun.

Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna *internet* dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari ha laman-halaman dokumen (*web*) itu. Proses yang dimulai dari permintaan *web client (browser)*, diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap – tiap proses. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web client* – nya.

(Khairil, Pemrograman *Web* (PHP dan *MySql* dengan *Dreamweaver* : 2)

Script untuk membuat *form login* (*input – an text* dan *password*) adalah sebagai berikut :

(Achmad Solichin, *Pemrograman Web dengan PHP dan MySql*, Jakarta, Universitas Budi Luhur : 35).

```
<html>

  <head><title>Login Here</title></head>

  <body>

    <FORM ACTION="proses05.php" METHOD="POST"
    NAME="input">

    <h2>Login Here...</h2>

    Username : <input type="text" name="username"><br>
    Password : <input type="password" name="password"><br>
    <input type="submit" name="Login" value="Login">
    <input type="reset" name="reset" value="Reset">

    </FORM>

  </body>

</html>
```

II.6. *MySql*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL *Database Management System* (DBMS) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus – kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti *Apache* yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu *MySQL AB*. *MySQL AB* memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL AB* adalah: *David Axmark*, *Allan Larsson*, dan *Michael "Monty" Widenius*. Beberapa kelebihan *MySQL* antara lain : *free* (bebas di – *download*), stabil dan tangguh, fleksibel dengan berbagai pemrograman, *security* yang baik, dukungan dari banyak komunitas, kemudahan *management database*, mendukung transaksi, perkembangan *software* yang cukup cepat.

(Achmad Solichin, *Pemrograman Web dengan PHP dan MySql*, Jakarta, Universitas Budi Luhur : 85).

Contoh *script* koneksi ke *database* Mysql (Achmad Solichin, Pemrograman *Web* dengan PHP dan MySQL, Jakarta, Universitas Budi Luhur : 114) sebagai berikut :

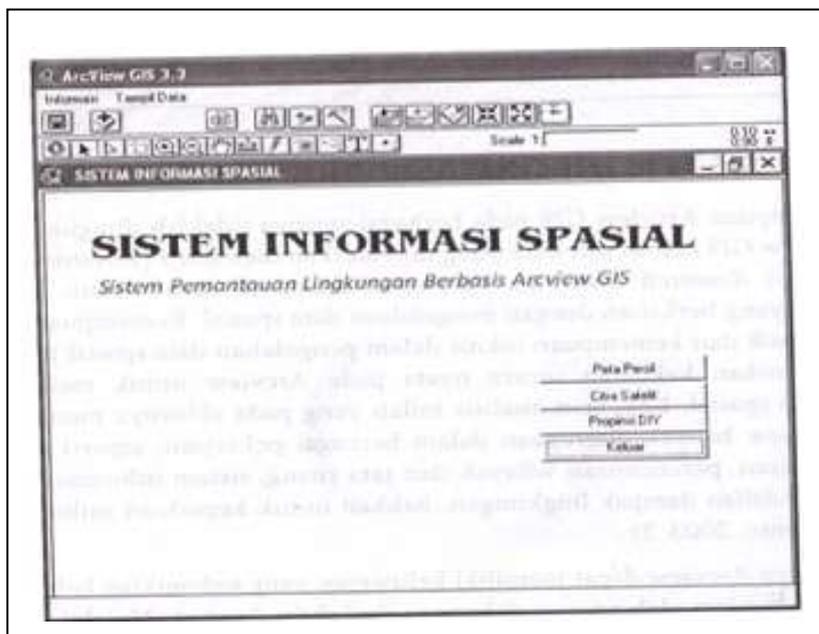
```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbnm = "pw2";
$conn = mysql_connect ($host, $user, $pass);
if ($conn) {
    $buka = mysql_select_db ($dbnm);
    if (!$buka) {
        die ("Database tidak dapat dibuka");
    }
} else {
    die ("Server MySQL tidak terhubung");
}
?>
```

II.7. *ArcView*

Kemampuan *Arcview* GIS pada berbagai serinya tidaklah diragukan lagi. *Arcview* GIS adalah software yang dikeluarkan 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 persis, pengendalian dampak lingkungan, bahkan untuk keperluan militer. Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *avenue*. Melalui *avenue* ini dapat dibentuk suatu “kemampuan baru” pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan spasial. *Avenue* dapat digunakan untuk “merombak” wajah *Arcview* sesuai kebutuhan penggunaannya.

Antarmuka sistem informasi (*interface*) dibentuk dengan memanfaatkan fasilitas *Customize* pada perangkat lunak *Arcview* GIS 3.3. Menu dan tombol dibentuk menggunakan teknik kustomasi tersebut. Teknik ini dipilih berdasarkan pada kemudahannya dalam membentuk menu dan berbagai tombol baru.

Adapun contoh Antarmuka Sistem Informasi Berbasis *Arcview* GIS dapat dilihat pada Gambar II.1 berikut :



Gambar II.1. Antarmuka Sistem Informasi Berbasis *Arcview* GIS
(Sumber : Eko Budiyanto ; 2010 : 178)

Dialog designer diperlukan untuk membentuk antarmuka penampil data atribut yang menjadi dasar pemilihan objek. *Dialog designer* yang dipilih adalah bentuk kotak daftar (*listbox*). Dengan menggunakan dialog ini operator akan memilih informasi apa yang akan dicari. Untuk menghubungkan menu dan tombol dengan berbagai aksi yang diinginkan maka perlu dibentuk skrip atau program. Skrip atau program ini dibentuk menggunakan bahasa Avenue. Setiap aksi yang diperlukan diuraikan menjadi baris-baris perintah pada skrip Avenue dan selanjutnya dikaitkan ke masing-masing menu atau tombol yang bersangkutan (Eko Budiyanto ; 2010 : 178).

II.8. *Unified Modelling Language*

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menentukan, visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. (Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 2).

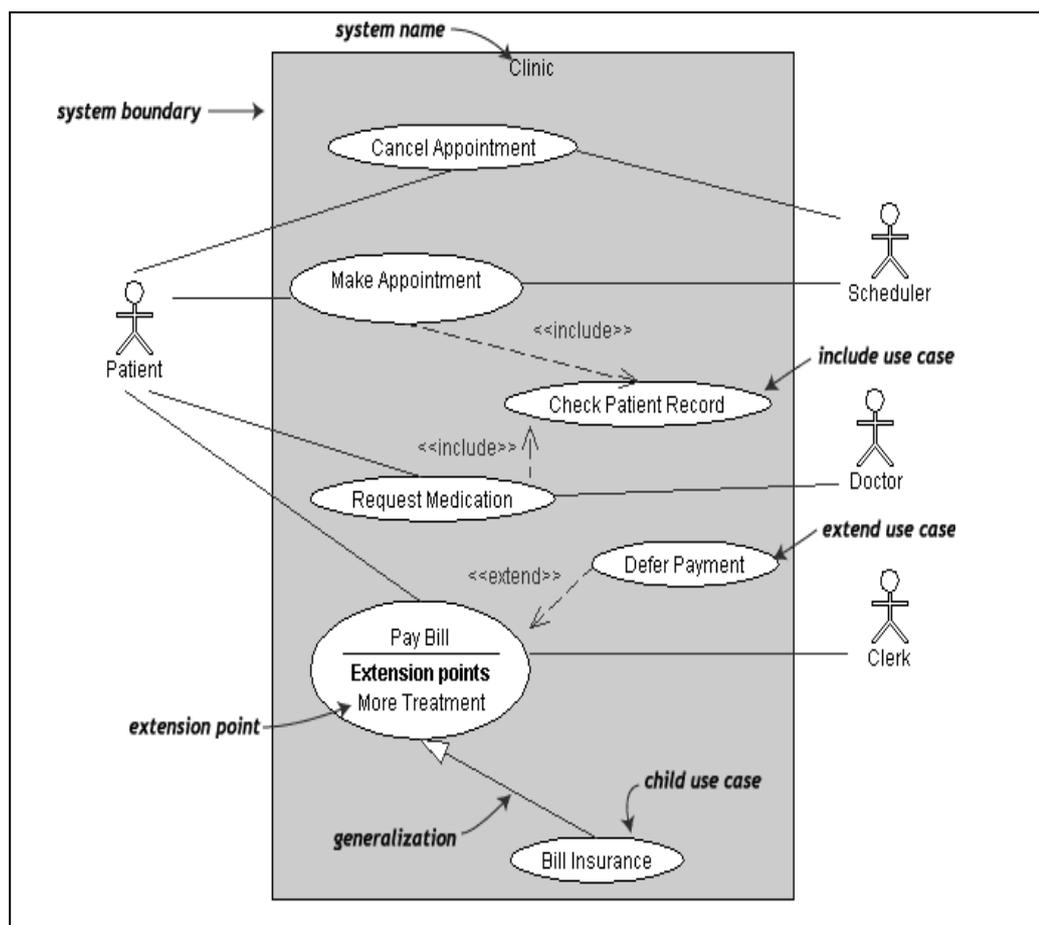
II.8.1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana” Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng – *create* sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng – *include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di – *include* akan dipanggil setiap kali *use case* yang meng – *include* dieksekusi secara normal. Sebuah *use case* dapat di – *include* oleh lebih dari satu *use case* lain, sehingga

duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng – *extend use case* lain dengan *behaviour* – nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. (Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 4).

Adapun contoh *Use Case Diagram* dapat dilihat pada Gambar II.2 berikut :



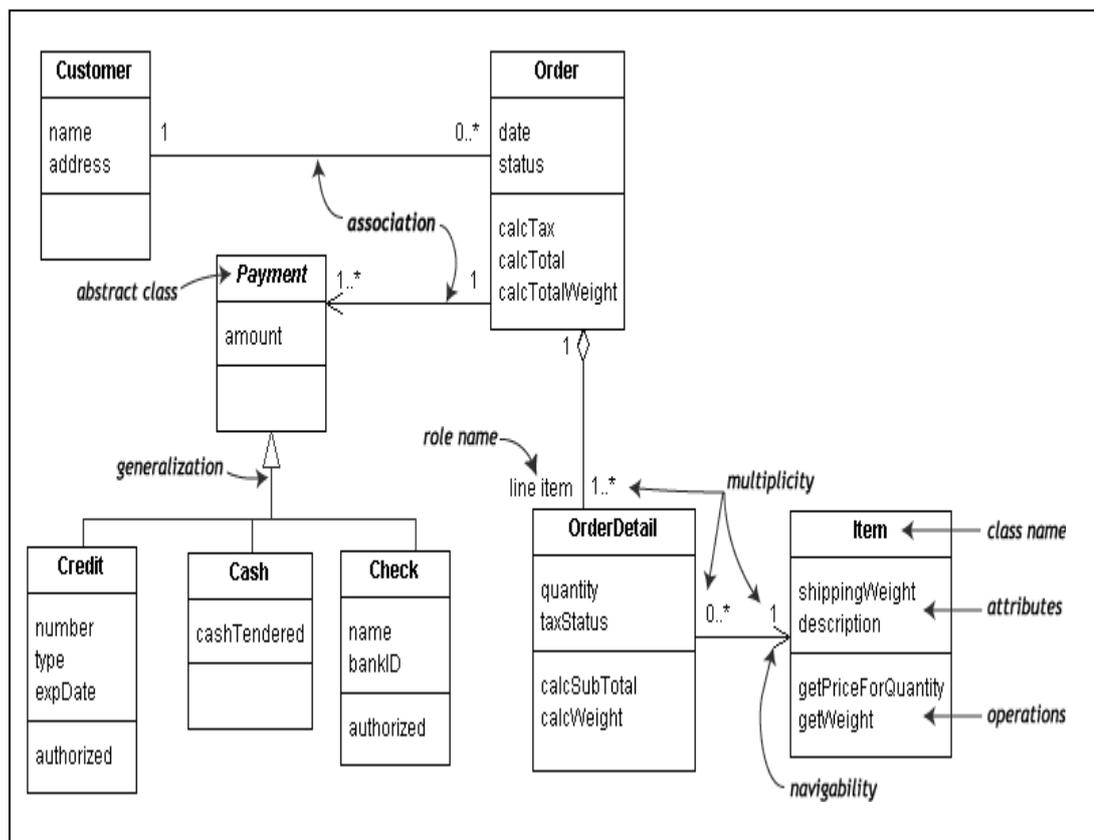
Gambar II.2. Contoh Use Case Diagram

(Sumber : Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 5)

II.8.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. (Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 5).

Adapun contoh Class Diagram dapat dilihat pada Gambar II.3 berikut :



Gambar II.3. Contoh Class Diagram

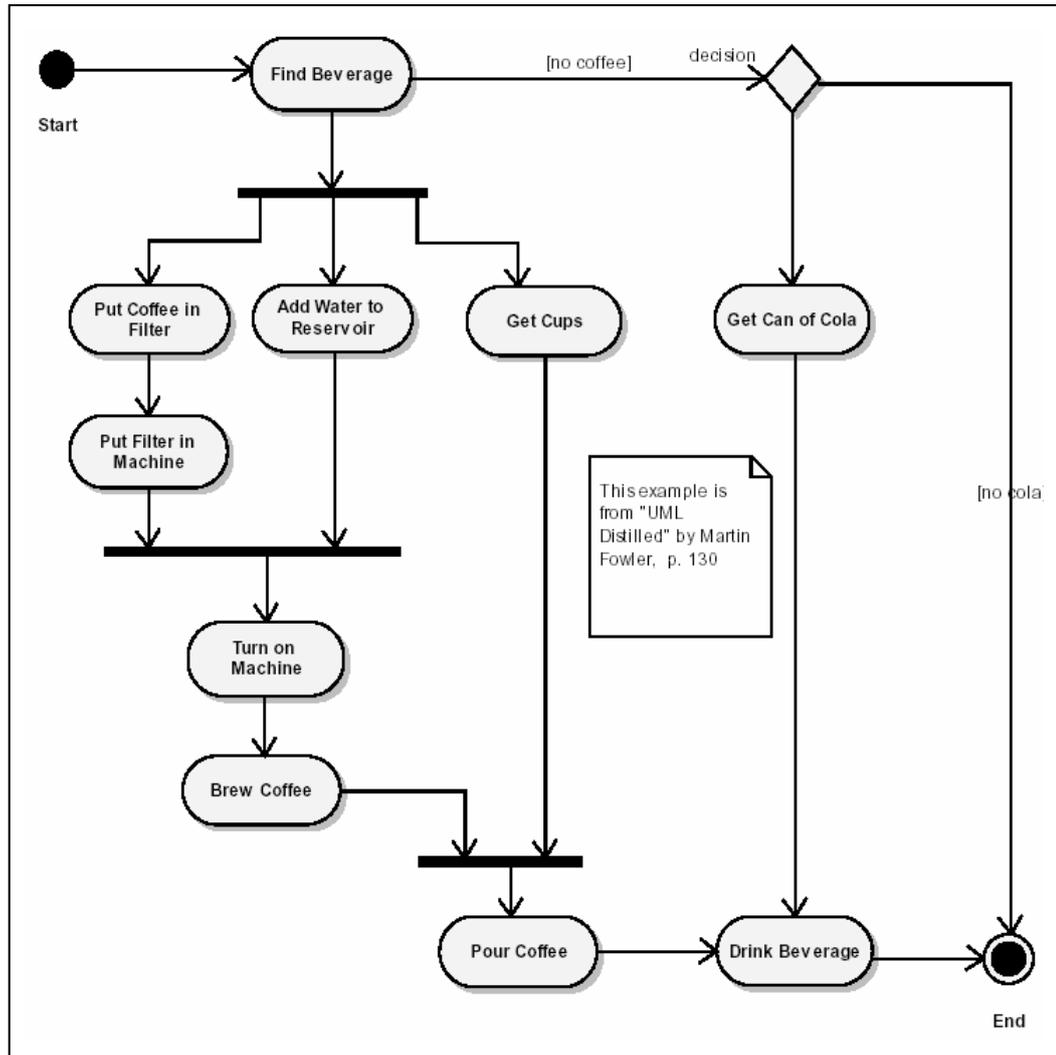
(Sumber : Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 6)

II.8.3. *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*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. (Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 7).

Adapun contoh *Activity Diagram* dapat dilihat pada Gambar II.4 berikut :



Gambar II.4. Contoh *Activity Diagram*

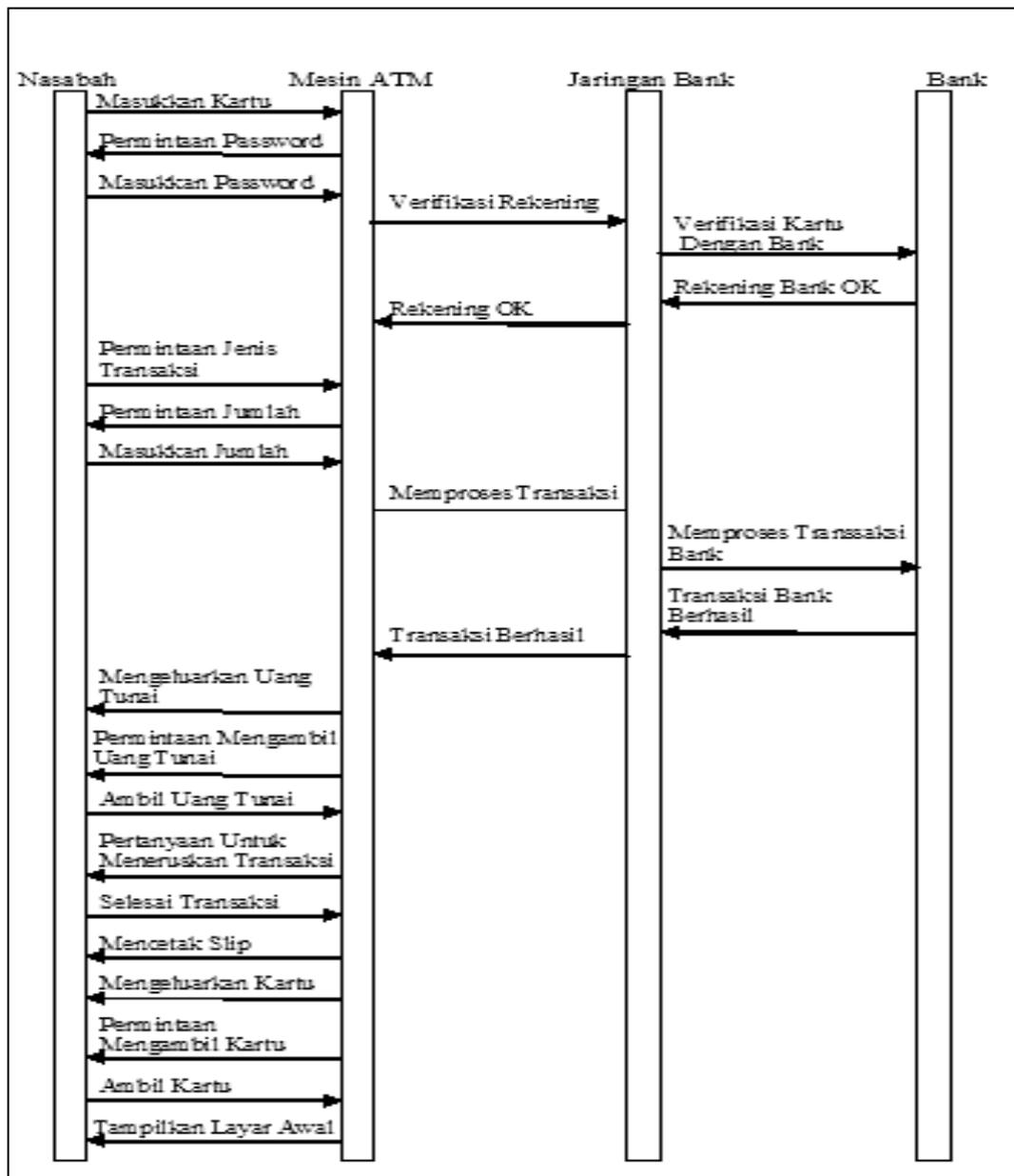
(Sumber : Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 8)

II.8.4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek – objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men – *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. (Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 8).

Adapun contoh *Sequence Diagram* dapat dilihat pada Gambar II.5 berikut :



Gambar II.5. Contoh *Sequence Diagram*

(Sumber : Sri Dharwiyanti dan Romi Satria Waryono ; 2008 : 8)

II.9. *Desain Database*

Proses dari mendesain struktur umum dari *database* terdiri dari :

1. *Desain logical*

Menentukan pada skema database, desain database membutuhkan yang kita namakan kumpulan yang baik dari skema hubungan.

2. Keputusan bisnis (*business decision*)

atribut apa yang seharusnya kita catat dalam *database*?

3. Keputusan ilmu komputer (*computer science decision*)

skema hubungan apa yang seharusnya kita punyai dan bagaimana seharusnya atribut di bagikan di antara skema hubungan yang bermacam-macam?

4. *Desain Physical*

Menentukan pada layout fisik dari database. (Arief Hidayat, Teori Perancangan Basis Data : 5).

II.9.1. Pengantar Normalisasi

Normalisasi adalah suatu proses yang di gunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga di peroleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut:

1. Mengandung redudansi sesedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan di hapus tanpa menimbulkan kesalahan atau ke tidak konsistenan.

Yang dimaksud redundansi data adalah data disimpan berkali-kali. Redundansi adalah istilah lain untuk duplikasi. Normalisasi juga dapat di pakai untuk memverifikasi relasi-relasi hasil tranformasi model E-R apakah sudah berstruktur baik atau belum. Normalisasi sendiri dilakukan melalui sejumlah.setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang di sebut bentuk normal adalah “suatu keadaan relasi yang di hasil kan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut”.

1. Bentuk norma pertama(1NF /*first normal form*).
2. Bentuk normal pertama(2NF/ *Secon normal form*).
3. Bentuk norma kedua(3NF / *Third normal form*).
4. Bentuk norma *Boyee-Codd(BCNF /Boye-Code normal form)*.
5. Bentuk norma keempat(4NF / *Fourth normal form*).
6. Bentuk norma kelima(5NF/ *fifth normal form*).(Abdul Kadir;2009:116)

II.9.2. Kamus Data

Database Management System (DBMS) memberikan fasilitas data *dictionary* (Kamus Data) untuk mendefinisikan nama – nama rinci data dan format penyimpanannya. Kamus data digunakan pada beberapa tahap yaitu pada tahap analisis sebagai alat komunikasi antar analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Selanjutnya pada

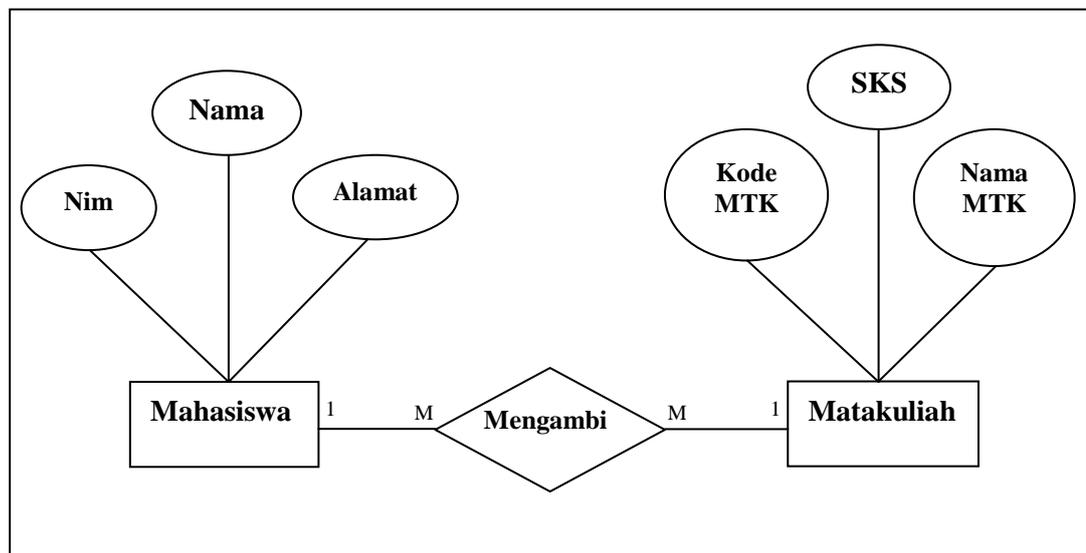
tahap perancangan sistem digunakan untuk merancang *input*, laporan – laporan, *database*. Kamus data berisi nama arus data, alias, bentuk data, arus data, penjelasan atau keterangan – keterangan, periode terjadinya transaksi, *volume* arus data yang mengalir dalam periode tertentu, dan struktur data. (Sistem Basis Data TFD2051 : 13).

II.9.3. Entity Relational Diagram (ERD)

Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data *semantik*, dan batasan konsistensi. Untuk mengilustrasikan konsep model data dapat disajikan dengan *entity relationship* model. *Entity relationship* mendeskripsikan rancangan basisdata pada tingkatan *logis*. *Entity relationship* (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek – objek dasar yang disebut entitas dan hubungan antar obyek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain.

Sebagai ilustrasi, bayangkan anda mengambil bagian sistem basis data universitas yang terdiri dari mahasiswa dan mata kuliah.

Adapun contoh ERD (*Entity Relational Diagram*) dapat dilihat pada Gambar II.6 berikut :



Gambar II.6. Menunjukkan ER Diagram
(Sumber : Janer Simarmata dan Iman paryudi ; 2006 : 60)

Diagram menunjukkan bahwa ada dua kumpulan entitas, yaitu mahasiswa dan mata kuliah, dan bahwa relasi mengambil mahasiswa dan mata kuliah (Janer Simarmata dan Iman Paryudi ; 2006 : 59)