

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau *variable* yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Selain itu suatu sistem tidak bisa lepas dari lingkungan sekitarnya maka umpan balik (*feed back*) dapat berasal dari lingkungan sistem yang dimaksud. (Tata Sutabri : 2012 : 10).

II.2. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Informasi dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya. (Tata Sutabri : 2012 : 29).

Kualitas informasi tergantung dari 3 hal yaitu :

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan – kesalahan dan tidak menyesatkan.

2. Tepat waktu (*Timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lain berbeda, misalnya informasi sebab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan dan akan lebih relevan bila ditunjukkan ahli teknik perusahaan. (Tata Sutabri : 2012 : 41).

II.3. Sistem Informasi

Sistem informasi adalah suatu sistem didalam 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 dengan laporan-laporan yang diperlukan (Tata Sutabri : 2012 : 46). Sistem informasi terdiri dari komponen – komponen yang disebut blok bangunan (*building block*) yaitu :

1. Blok masukan (input block)

Input mewakili data yang masuk kedalam sistem informasi, Input disini 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 semua 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, teknologi terdiri dari 3 bagian utama, yaitu teknis (brainware), Perangkat lunak (software), perangkat keras (hardware).

5. Blok basis data (*database block*)

Basis data atau database merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan – kecurangan, kegagalan – kegagalan. (Tata Sutabri : 2012 : 47-48).

II.4. Sistem Informasi Geografis

Sistem informasi geografis dirancang untuk mengumpulkan, menyimpan dan menganalisis obyek dimana lokasi geografis merupakan karakteristik yang penting. Sistem informasi geografis hingga saat ini merupakan sistem yang sangat menarik. Menurut Prahasta (2006 : 1), Sistem ini dapat mengintegrasikan data spasial (peta vektor dan citra digital), atribut (tabel sistem basis data) serta *properties* penting lainnya. Kemampuan tersebutlah yang membedakan sistem informasi geografis dengan sistem informasi lain dan membuat sistem informasi geografis lebih bermanfaat dalam memberikan informasi yang mendekati kondisi dunia nyata, memprediksi suatu hasil dan perencanaan strategis.

Menurut Prahasta (2006 : 2), Fungsi perangkat lunak sistem informasi geografis yang paling utama setelah sebagai perangkat lunak *mapping system* dengan kemampuan kartografisnya adalah kemampuannya dalam menjawab hal-hal yang terkait analisis (*query*). Sistem informasi geografis dapat memecahkan masalah-masalah analisis spasial, atribut dan kombinasinya. Dengan memanfaatkan sistem informasi geografis, setiap pengguna dapat melakukan proses-proses analisis dan pembuatan peta (kartografis) digital secara mudah. Selain itu, pada saat ini sistem informasi geografis juga dilengkapi dengan kemampuan menampilkan dan mengolah data permukaan tiga dimensi (*raster grid*, DTM/DEM) sebagai alat bantu pemodelan dengan aspek dimensi ketiga. (Jurnal Wartika:2010:5-6).

Dengan demikian, GIS diharapkan mampu memberikan kemudahan-kemudahan yang diinginkan yaitu:

1. Penanganan data geospasial menjadi lebih baik dalam format baku.
2. Revisi dan pemutakhiran data menjadi lebih muda.
3. Data geospasial dan informasi menjadi lebih mudah dicari, dianalisa dan direpresentasikan.
4. Menjadi produk yang mempunyai nilai tambah.
5. Kemampuan menukar data geospasial.
6. Penghematan waktu dan biaya.
7. keputusan yang diambil menjadi lebih baik.

Komponen pada Geographical Information System terdiri atas :

1. *Hardware*

GIS membutuhkan komputer untuk penyimpanan dan pemrosesan data. Ukuran dari sistem komputerisasi bergantung pada tipe GIS itu sendiri. GIS dengan skala yang kecil hanya membutuhkan PC (*personal computer*) yang kecil dan sebaliknya. Ketika GIS yang 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 prosesor yang cepat. Untuk mengubah peta ke dalam bentuk digital diperlukan hardware yang disebut digitizer.

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:

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

Inti dari software GIS adalah software GIS itu sendiri yang mampu menyediakan fungsi-fungsi untuk penyimpanan, pengaturan, link, query dan analisa data geografi. Beberapa contoh software GIS adalah ArcView, MapInfo, ArcInfo untuk SIG; CAD system untuk entry graphic data; dan ERDAS serta ER-MAP untuk proses remote sensing data. Modul dasar perangkat lunak SIG: modul pemasukan dan pembetulan data, modul penyimpanan dan pengorganisasian data, modul pemrosesan dan penyajian data, modul transformasi data, modul interaksi dengan pengguna (*input query*).

3. Data

SIG merupakan perangkat pengelolaan basis data (*DBMS = Data Base Management System*) dimana interaksi dengan pemakai dilakukan dengan suatu sistem antar muka dan sistem query dan basis data dibangun untuk aplikasi multiuser. SIG merupakan perangkat analisis keruangan (*spatial analysis*) dengan kelebihan dapat mengelola data spasial dan data non-spasial sekaligus. (Jurnal Rahmat Husein:2006:4-6).

II.5. *PHP*

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa script server-side dalam pengembangan web yang disisipkan pada dokumen HTML.

Penggunaan *PHP* memungkinkan web dapat dibuat dinamis sehingga maintenance situs web tersebut menjadi lebih mudah dan efisien. *PHP* merupakan software *Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di download secara bebas dari situs resminya. Salah satu fitur yang dapat diandalkan oleh *PHP* adalah dukungannya terhadap banyak *database*, seperti *Adabas D*, *dBase*, *Direct MS-SQL*, *Empress*, *FilePro*, *FrontBase*, *Hyperware*, *IBM DB2*, *Informix*, *MSQL*, *MySql* dan masih banyak lagi (Dadan Sutisna : 2008 : 40).

Bagi pemula belajar *PHP*, istilah ini sangat penting untuk mendukung dalam pembuatan program, tetapi bagi yang telah belajar bahasa pemrograman seperti bahasa C, *Pascal* dan *Java*, istilah ini sudah sering didengar. Kita Perlu mengingat perbedaan tipe data, variabel, konstanta dan operator yang terdapat dalam *PHP*. Tipe data merupakan jenis data, setiap data pasti memiliki jenis data tertentu. Jenis data berguna untuk menghindari ekspresi yang tipe datanya bercampur. Sebagai contoh '*PHP*' + 4.0 (jenis data *string PHP* mau ditambah dengan tipe data *numerik* 4.0). *PHP* mengenal 3 kelompok jenis data yaitu :

1. Integer (bilangan bulat)

Tipe data integer merupakan tipe data yang direpresentasikan berbentuk angka dan tidak memakai tanda titik desimal. Contoh : \$angka=2-;

2. Tipe data Double (bilangan pecahan)

Tipe data double merupakan tipe data yang direpresentasikan memakai tanda titik desimal. Contoh : \$angka=20.50;

3. Tipe data String (teks)

Tipe data string merupakan tipe data yang direpresentasikan diapit tanda petik tunggal (' ') atau petik ganda (" "). Contoh : \$kantor="CV.Noornet";
(Yuniar Supardi :2010 : 17-18).

II.5.1. Pemrograman Berorientasi Objek dalam PHP

PHP pada awalnya hanyalah kumpulan script sederhana. Dalam perkembangannya, selanjutnya ditambahkan berbagai fitur pemrograman berorientasi objek. Hal ini dimulai sejak PHP 4. Dengan lahirnya PHP 5, fitur-fitur pemrograman berorientasi objek semakin mantap dan semakin cepat. Dengan PHP 5, script yang menggunakan konsep object-oriented akan lebih cepat dan lebih efisien. Pemrograman berorientasi objek atau *object-oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir setiap perguruan tinggi di dunia mengajarkan konsep OOP ini pada mahasiswanya. Pemrograman yang banyak dipakai dalam penerapan konsep OOP adalah Java dan C++. OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah object. Sebuah

object mewakili suatu bagian program yang akan diselesaikan. Beberapa konsep OOP dasar, antara lain :

1. *Encapsulation (Class dan Object)*
2. *Inheritance* (Penurunan sifat), dan
3. *Polymorphisme*

PHP khususnya PHP 5 sudah mendukung beberapa konsep OOP. Akan tetapi PHP 5 tidak mendukung konsep *Multiple-inheritance* dan *polymorphisme*.

File PHP pada server ditulis dengan mengimbuahkan ekstensi .php (misalnya nama-file.php). Server kemudian memeriksa skrip yang ada pada file tersebut, jika didalamnya terdapat perintah PHP, server akan menerjemahkannya sebelum masuk ke browser. Berikut beberapa contoh untuk memasukkan skrip PHP.(Dadan Sutisna : 2008 : 41):

1. Penulisan PHP dimulai dengan <? Dan diakhiri dengan ?>

```
</
```

```
Disini skrip PHP
```

```
?>
```

2. Setiap perintah pada PHP diakhiri dengan titik koma (;)

```
<?
```

```
Perintah 1;
```

```
Perintah 2;
```

```
Dst;
```

```
?>
```

3. Variabel PHP ditulis dengan tanda dollar (\$) dan kemudian diikuti nama variable tanpa spasi.

```
<?
```

```
$namavariabel = "ini adalah variable";
```

```
$hitung = 2 + 5;
```

```
?> (Dadan Sutisna : 2008 : 41)
```

II.5.2 ArcView

ArcView adalah salah satu *software* pengolah Sistem Informasi Geografik (SIG/GIS). Sistem Informasi Geografik sendiri merupakan suatu sistem yang dirancang untuk menyimpan, memanipulasi, menganalisis, dan menyajikan informasi geografi. Mungkin anda sudah kenal kenal dengan yang namanya peta. Perlu diketahui bahwa peta juga bisa disebut *SIG* atau istilahnya *SIG* Konvensional. Terdapat beberapa perbedaan antara peta di atas kertas (peta analog) dan *SIG* yang berbasis komputer. Perbedaannya adalah bahwa peta menampilkan data secara grafis tanpa melibatkan basis data. Sedangkan *SIG* adalah suatu sistem yang melibatkan peta dan basis data. Dengan kata lain peta adalah bagian dari *SIG*. Sedangkan pada *ArcView* anda dapat melakukan beberapa hal yang peta biasa tidak dapat melakukannya.

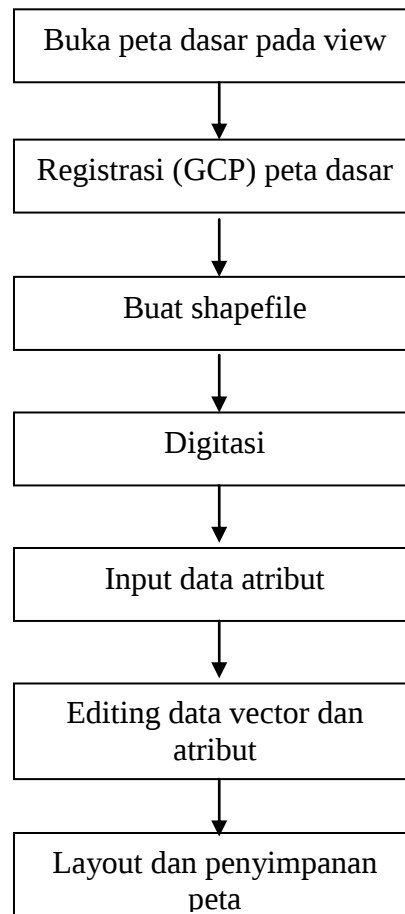
Shapefile adalah *file* yang menyimpan data *vector* dalam *arcview*. *Shapefile* inilah yang kemudian diolah dan dianalisis dalam berbagai pekerjaan spasial dengan *arcview*. Saat ditampilkan dalam lembar *view*, *shapefile* masih sebagai sebuah *theme*. Data *vector* yang tidak disimpan dalam sebuah *shapefile*

hanya akan menjadi objek grafik dalam lembar *view*. Objek grafik tersebut akan hilang jika dimasukkan dalam *shapefile*. Oleh karena itu, sebelum proses *digitasi* dilakukan, terlebih dahulu harus dibentuk sebuah *shapefile* kosong untuk wadah data *vector* yang telah ter*digitasi*. (Eko Budiyo :2010 : 1).

Digitasi merupakan proses pembentukan data *vector*. Dalam sistem informasi geografis dan pemetaan digital, data *vector* banyak digunakan sebagai dasar analisis dan berbagai proses. *Digitasi* yang dilakukan tanpa disimpan pada sebuah *theme (file .shp)*, maka akan menjadi bentuk grafis. Data grafis ini juga merupakan data *vector* tetapi tidak memiliki basis data *atributal*. Dalam pembuatan peta *digital*, data grafis harus disimpan di dalam sebuah *shapefile (file .shp)*. Oleh karena itu, proses *digitasi* didahului dengan pembuatan sebuah *shapefile* kosong. (Eko Budiyo :2010 : 19).

Peta dasar adalah peta yang akan digunakan sebagai referensi pembuatan peta. Peta dasar ini dapat berupa data citra satelit, data foto udara, atau data *raster* peta hasil proses penyiaman (*scanning*). Proses *updating* peta sering menggunakan data penginderaan jauh seperti citra satelit resolusi tinggi atau foto udara sebagai peta dasarnya. Sementara itu proses reproduksi peta biasanya menggunakan data *raster* peta tertentu yang disiam. Teknik *digitasi* peta dasar pada masa lalu sering menggunakan alat bantu yang bernama *digitizer*. Dalam perkembangannya, proses *digitasi* peta dasar kemudian dilakukan secara *onscreen*. Teknik *digitasi onscreen* ini adalah teknik *digitasi* peta dasar secara langsung pada layar computer. Akurasi teknik *digitasi* peta dasar *onscreen relative* lebih tinggi karena adanya kemampuan *zoom* pada setiap perangkat lunak yang

digunakan untuk *digitasi*. (Eko Budiyanto :2010 : 25). Secara garis besar, pembuatan peta *digital* dilakukan dalam rangkaian proses berikut :



**Gambar II.1. Proses Pembuatan Peta Digital
(Sumber Eko Budiyanto :2010 : 26)**

II.6. Alat Bantu Pengembangan Sistem

Alat bantu pengembangan sistem yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

II.6.1 UML (Unified Modelling Language)

Unified Modelling Language (UML) merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem, mau tidak mau pasti akan menjumpai *UML*, baik kita sendiri yang membuat atau sekedar membaca diagram *UML* buatan orang lain. (Prabowo Pudjo Widodo ; 2011 : 7). Sebelum ada *UML*, para pengembang bahasa pemrograman berorientasi objek sulit untuk berkomunikasi satu sama lain. Ada kira-kira 50 jenis notasi dan grafik yang menggambarkan bahasa pemrograman berorientasi objek pada waktu itu. Para pengguna notasi yang berlainan ini saling berebut pengaruh agar notasi yang mereka gunakan jadi notasi standar. (Prabowo Pudjo Widodo ; 2011 : 8).

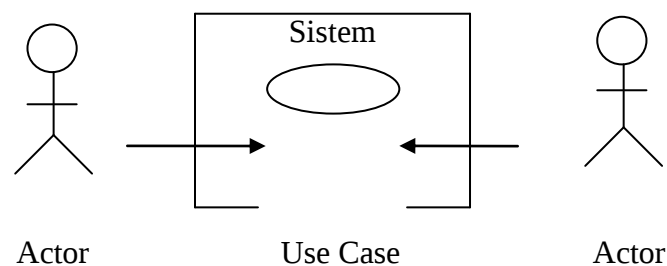
Sejak tahun 1997, divisi *Revision Task Force (RTF)* milik *OMG* beberapa kali merevisi *UML*. Revisi dimaksudkan untuk memperkuat konsistensi notasi, meningkatkan kekompakan antara *user* dan pengembang perangkat lunak. Akan tetapi *UML* terpaksa mengikuti perkembangan *software-software* berbasis objek yang ada. Setelah dilakukan perubahan secara sistematis, akhirnya dihasilkan *UML 2.0* pada tahun 2003. (Prabowo Pudjo Widodo ; 2011 : 9).

II.6.2 Diagram – diagram Pada Metode UML

1. Use Case Diagram

Use case adalah alat bantu terbaik guna menstimulasikan pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka

bermaksud menggunakan sebuah sistem. Ide dasarnya adalah bagaimana melibatkan penggunaan sistem di fase – fase awal analisis dan perancangan sistem. Diagram use case menunjukkan 3 aspek dari sistem yaitu actor , use case dan sistem / sub sistem boundary. Actor mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case . Gambar II.4 mengilustrasikan actor , use case dan boundary.

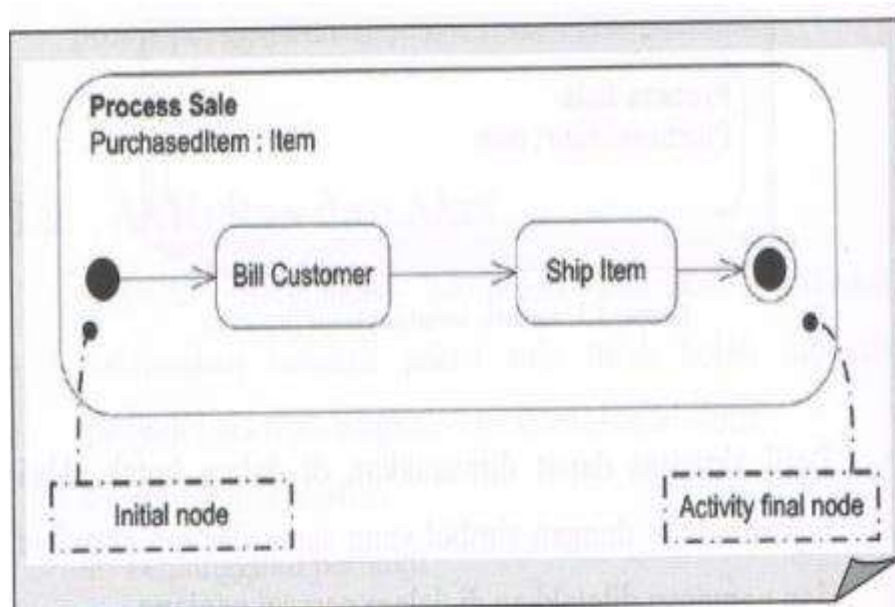


Gambar II.2 Use Case Model

Sumber : (Prabowo Pudjo Widodo ; 2011 : 17)

2. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut gambar dari sederhana dari *activity diagram*.

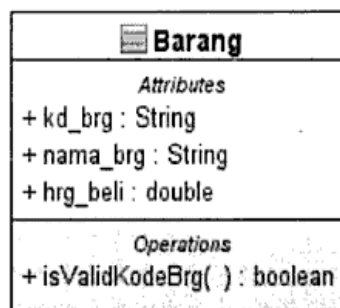


Gambar II.3 Contoh Activity Diagram Sederhana

Sumber : (Prabowo Pudjo Widodo ; 2011 : 143)

3. Class Diagram

Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.(Prabowo Pudjo widodo ; 2011 :10)



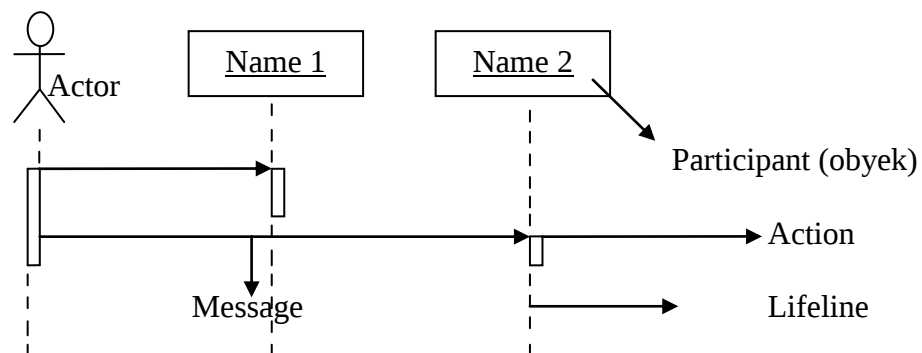
Gambar II.4 Contoh Class Diagram

Sumber : (Prabowo Pudjo Widodo ; 2011:41)

4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakkan diantara obyek – obyek di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*



Gambar II.5 Simbol – simbol yang ada pada sequence diagram

Sumber : (Prabowo Pudjo Widodo ; 2011:104)

II.7. Sistem Basis Data

Basis adalah markas, gudang, tempat bersarang tau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (dosen, mahasiswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Basis data adalah kumpulan data yang saling berhubungan yang disimpan/diorganisasi secara bersama, dalam bentuk sedemikian rupa, dan tanpa redudansi (pengulangan) yang tidak perlu

supaya dapat dimanfaatkan kembali dengan cepat dan mudah untuk memenuhi berbagai kebutuhan. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 3).

Komponen-komponen pada sebuah sistem basis data antara lain:

1. Data
2. Perangkat keras
3. Sistem operasi
4. Basis data
5. DBMS (*Database Management System*)
6. Pemakai
7. Aplikasi lain. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 4).

DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. DBMS merupakan antarmuka pengguna basis data (baik merupakan pengguna langsung maupun aplikasi) dengan data yang tersimpan. Penyimpanan data oleh DBMS disesuaikan dengan bentuk model datanya, beberapa contoh DBMS adalah *PostgreSQL*, *MYSQL*, *DB2*, *Oracle*, *SQLserver* dan lain lain. Sebelum ada DBMS, data pada umumnya disimpan dalam bentuk *flat file*, yaitu *file teks* yang ada pada sistem operasi. Sampai sekarang pun masih banyak aplikasi yang menyimpan dalam bentuk *flat file* secara langsung. Contoh yang sederhana adalah misalnya dalam sistem operasi linux terdapat *file password* yang digunakan untuk menyimpan nama penggunaanya. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 5).

II.8 MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu, ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya).

Database digunakan untuk penyimpanan data, demikian pula dengan *MYSQL*. Kita akan memanggil data pada *MYSQL* melalui *PHP*, kemudian hasilnya dikirim kekomputer *klien* untuk ditampilkan pada *browser*. Data pada *MYSQL* dapat dipanggil, dihapus atau di tambah melalui *query* .(Dadan Sutisna : 2008 : 46)

Langkah untuk menjalankan *webserver local* adalah :

1. Buka program *browser*, misalnya ***Mozilla Firefox*** atau ***Internet Explorer***.
2. Dalam jendela *browser*, ketil ***localhost*** dibagian ***Address***, kemudian tekan ***Enter***.
3. Berikutnya adalah membuka halaman *phpMyAdmin*, klik link ***phpMyAdmin Database Manager Version 2.10.2***.
4. Berikutnya program akan meminta memasukkan *user* dan *password* dari server *MySQL*. Ketik *user* nama ***root*** pada kotak ***User Name***, dan masukkan *password*-nya pada kotak ***Password*** sesuai dengan *password* pada saat penginstalan.

II.9 Konsep DataBase

Database adalah kumpulan beragam elemen informasi yang akan digunakan demi tujuan klasifikasi. Terdapat tiga tingkat arsitektur yang terkait dengan *database* dan sistem manajemen *database* yaitu tingkat konseptual, tingkat logika, dan tingkat fisik.

Dalam konsep perancangan *database* terdapat pendekatan yang perlu diperhatikan yaitu sebagai berikut :

II.9.1 Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel yang menunjukkan entitas sekaligus relasinya. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 40).

Tahap normalisasi terdiri dari beberapa bentuk :

1. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal Kesatu (1NF/ *First Normal Form*)

Bentuk normal kesatu mempunyai ciri : setiap data dibentuk dalam *file file* (*file* datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa "atomic value". Tidak ada set atribut yang berulang atau atribut bernilai ganda (*multivalue*). Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukan pecahan kata sehingga artinya lain. Atom adalah zat terkecil

yang masih memiliki sifat induknya, bila dipecah lagi, maka ia tidak memiliki sifat induknya.

3. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Bentuk normal kedua memiliki syarat : bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung fungsi pada kunci utama/*primary key* sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya. Tahap normalisasi terdiri dari beberapa bentuk:

4. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Untuk menjadi bentuk normal ketiga, relasi haruslah dalam bentuk normal kedua dan semua atribut dalam primer tidak punya hubungan yang transif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh. Contoh pada bentuk kedua di atas termasuk juga bentuk normal ketiga seluruh atribut yang ada disitu bergantung penuh pada kunci primernya.

5. Bentuk Normal *Boyce Codd* (BCNF/ *Boyce Codd Normal Form*)

Boyce Codd Normal Form mempunyai paksaan yang lebih kuat dari pada bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey*. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 40).

II.9.2 Kamus Data

Kamus data ikut berperan dalam perancangan dan pembanguna SI karena peralatan berfungsi untuk :

1. Menjelaskan arti aliran data dan penyimpan dalam penggambaran data data flow diagram.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran, misalnya data alamat diurai menjadi nama jalan,nomor,kota,negara dan kode pos.
3. Menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

Tabel II.1 Simbol-simbol dalam kamus data

Simbol	Uraian
=	Terdiri atas,mendefinisikan,diuraikan menjadi,artinya Contoh : Nama=Sebutan + Nama1+Nama2 + Gelar1 + Gelar2
+	Dan
()	Optional (pilihan boleh ada atau boleh tidak) Contoh : Alamat = Alamat rumah + (Alamat surat)
{ }	Pengulangan Contoh : Nama1 = {Karakter _valid}
[]	Memilih salah satu dari sejumlah alternatif,seleksi. Contoh : Sebutan =[Bapak Ibu Yang Mulia]
**	Komentar Contoh : *Seminar yang akan diikuti*

	Pemisah sejumlah alternatif pilihan antara simbol []
--	---

Sumber : (Budi Sutedjo Dharma Oetomo ; 2006 : 118-119)

II.10 E-R Diagram (ERD)

ERD merupakan notasi grafis dalam pemodelan data *konseptual* yang mendeskripsikan hubungan antar penyimpanan. *ERD* juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lainnya dalam dunia nyata. (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 18)

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar dua data. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. Entity

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh adalah barang, pemasok, pekerja dan lain-lain.

Seandainya adalah A maka barang A adalah isi dari barang, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan. Karena itu harus dibedakan antara entitas sebagai bentuk umum dari deskripsi tertentu dan isi entitas seperti A dan B dalam contoh diatas. Entitas dapat digambarkan dalam bentuk persegi empat.

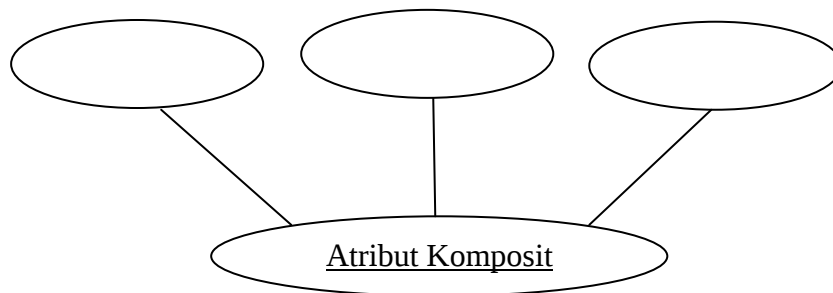


Gambar II.6. Entitas

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 19)

2. Atribut

Entitas mempunyai elemen yang disebut *atribut* dan berfungsi mendeskripsikan karakter *entitas*, misalnya atribut nama barang dari *entitas* barang. Setiap *ERD* bisa berisi lebih dari satu *atribut*. Penggambaran atribut dalam *ERD* sering hanya digambar berupa elips untuk semua jenis atribut dan dengan garis bawah untuk atribut sebagai kunci.

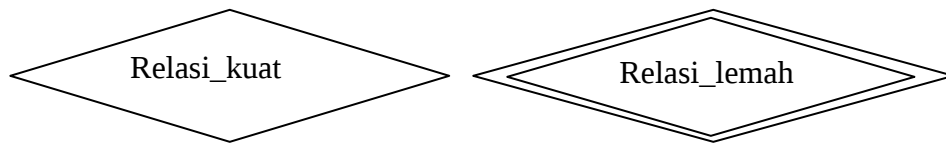


Gambar II.7 Atribut

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 23)

3. Hubungan / *relationship*

Belah ketupat merupakan penggambaran hubungan (relasi) antarentitas atau sering disebut kerelasian. Ada dua macam penggambaran relasi yaitu relasi kuat dan relasi lemah. Relasi kuat biasanya menghubungkan antarentitas kuat, sedangkan relasi lemah untuk menghubungkan antara entitas kuat dengan entitas lemah. *Relationship* digambarkan sebagai berikut :



Gambar II.8. Relationship

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

Untuk menghubungkan entitas-kerelasiaan-entitas digunakan garis lurus seperti gambar berikut ini :

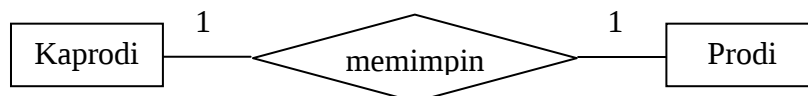


Gambar II.9. Kerelasiaan antarentitas

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

Jenis-jenis hubungan:

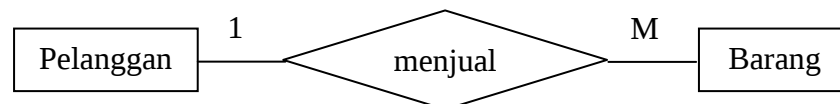
- a. Satu ke satu, misalnya suatu perusahaan mempunyai aturan satu supir hanya boleh menangani satu kendaraan karena alasan tertentu.



Gambar II.10. Relational 1 to 1

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

- b. Satu ke banyak atau banyak ke satu, misalnya suatu perusahaan selalu berasumsi bahwa satu pelanggan dapat membeli banyak barang



Gambar II.11. Relational 1 to Many

Sumber : (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 25)