BAB II

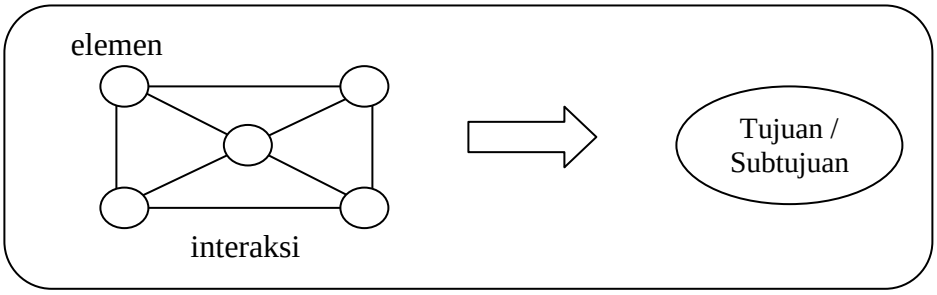
TINJAUAN PUSTAKA

II.1. Sistem

II.1.1. Pengertian Sistem

Sistem adalah suatu usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antarbagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerjasama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu dikembangkan dan dikendalikan (Marimin, dkk, 2008: 1).

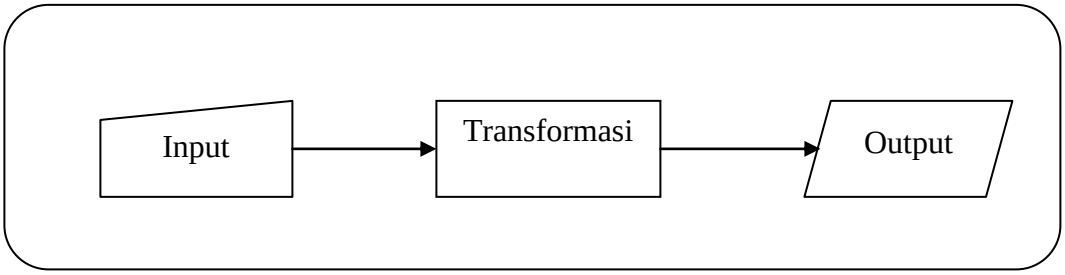
Definisi tersebut menunjukkan bahwa sistem sebagai gugus dari elemen-elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan. Pengertian sistem secara skematis dapat dilihat pada Gambar II.1 berikut.



Gambar II. 1: Pengertian Sistem
Sumber : Marimin, dkk (2007: 2)

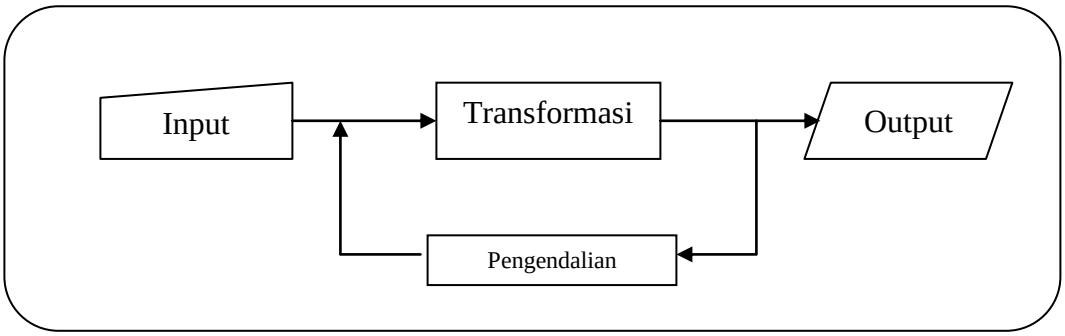
Sifat-sifat dasar dari suatu sistem antara lain adalah:

1. Pencapaian tujuan, orientasi pencapaian tujuan akan memberikan sifat dinamis kepada sistem, memberi ciri perubahan yang terus menerus dalam usaha mencapai tujuan.
2. Kesatuan usaha, mencerminkan suatu sifat dasar dari sistem, dimana hasil keseluruhan melebihi dari jumlah bagian-bagiannya atau sering disebut konsep sinergi.
3. Keterbukaan terhadap lingkungan, lingkungan merupakan sumber kesempatan maupun hambatan pengembangan. Keterbukaan terhadap lingkungan membuat penilaian terhadap suatu sistem menjadi relatif atau yang dinamakan *equifinality* atau pencapaian tujuan suatu sistem tidak mutlak harus dilakukan dengan satu cara terbaik. Tetapi pencapaian tujuan suatu sistem dapat dilakukan melalui berbagai cara sesuai dengan tantangan lingkungan yang dihadapi.
4. Transformasi, merupakan proses perubahan *input* menjadi *output* yang dilakukan oleh sistem. Proses transformasi di ilustrasikan seperti pada Gambar II.2 berikut di bawah ini.



Gambar II. 2: Proses Transformasi *Input* Menjadi *Output*
Sumber : Mariamin, dkk (2008: 2).

5. Hubungan antarbagian, kaitan antara subsistem inilah yang akan memberikan analisis sistem, suatu dasar pemahaman yang lebih luas.
6. Sistem ada berbagai macam, antara lain sistem terbuka, sistem tertutup, dan sistem dengan umpan balik.
7. Mekanisme pengendalian, mekanisme ini menyangkut sistem umpan balik yang merupakan suatu bagian yang memberi informasi kepada sistem mengenai efek dari perilaku sistem terhadap pencapaian tujuan atau pemecahan persoalan yang dihadapi (Marimin, dkk, 2008: 2). Skema proses transformasi sistem dengan mekanisme pengendalian disajikan pada Gambar II.3 berikut di bawah ini.



Gambar II. 3: Skema Proses Transformasi Sistem dengan Mekanisme Pengendalian

Sumber : Mariamin, dkk (2008: 3).

II.1.2. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya :

1. Batasan (*boundary*), penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*), segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala dan *input* terhadap suatu sistem.
3. Masukan (*input*), sumber daya atau produk (informasi, laporan, dokumen, tampilan layer komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
4. Komponen (*component*), kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan *input* menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
5. Penghubung (*interface*), tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
6. Penyimpanan (*storage*), area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (Hanif al Fatta, 2007: 5-6).

II.1.3. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, diantaranya adalah seperti yang diuraikan berikut :

1. Tujuan
2. Masukan
3. Keluaran
4. Proses
5. Mekanisme pengendalian, dan
6. Umpan balik

1. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem lainnya berbeda-beda. Begitu pula yang berlaku pada sistem informasi. Setiap sistem informasi memiliki suatu tujuan, tetapi dengan tujuan yang berbeda-beda. Walaupun demikian, tujuan utama yang umum menurut Hall (2000) ada tiga macam yaitu:

a. Untuk mendukung fungsi kepengurusan manajemen
b. Untuk mendukung pengambilan keputusan manajemen
c. Untuk mendukung kegiatan operasi perusahaan.

Secara lebih spesifik, tujuan sistem informasi bergantung pada kegiatan yang ditangani. Namun kecenderungan penggunaan sistem informasi lebih ditujukan pada usaha menuju keunggulan kompetitit, yang artinya mampu bersaing dan mengungguli pesaing (Abdul Kadir, 2003: 54-57).

2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa dari pelanggan). Pada sistem informasi, masukan dapat berupa data transaksi, dan data non transaksi (misalnya surat pemberitahuan), serta instruksi (Abdul Kadir, 2003: 54-57).

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembangunan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien. Pada sistem informasi, proses dapat berupa suatu tindakan yang bermacam-macam. Meringkas data, melakukan perhitungan dan mengurutkan data merupakan beberapa contoh proses (Abdul Kadir, 2003: 54-57).

4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem intodamasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan dan lain sebagainya (Abdul Kadir, 2003: 54-57).

5. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (control mechanism) diwujudkan dengan menggunakan umpan balik (feedback), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan. Dalam bentuk yang sederhana, dilakukan perbandingan antara keluaran sistem dan keluaran yang dikehendaki (standar). Jika terdapat penyimpangan maka akan dilakukan pengiriman masukan untuk melakukan penyesuaian terhadap proses supaya keluaran berikutnya mendekati standart. Bila penyebab penyimpangan terletak pada proses, maka prosesnyalah yang diperbaiki. Pada sistem informasi, cara yang pertama dapat memberikan masukan pada setiap infividu atau emberikan ringkasan kinerja terakhir untuk kegiatan manajemen. Adapun hal yang serting terjadi pada sistem informasi karena program komputernyalah yang salah atau keluarannya dikehendaki untuk diubah (Abdul Kadir, 2003: 54-57).

Umpan balik seperti yang diutarakan di depan, yaitu menyesuaikan penyimpangan terhadap standar biasa disebut umpan balik negatif (negative feedback). Contoh penerapan umpan balik negatif yaitu penerapan thermostat pada sistem pendingin (AC). Alat inilah yang berfungsi untuk mengontrol agar agar suhu

ruangan sesuai dengan yang diinginkan pemakai. Pengontrolan suhu dilakukan dengan menggunakan sensor (Abdul Kadir, 2003: 54-57).

II.2. Informasi

II.2.1. Pengertian Informasi

McFadden, dkk (1999) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data teresebut. Shannon dan Weaver, dua orang insinyur listrik, melakukan pendekatan secara matematis untuk mendefinisikan informasi (Kroenke, 1992). Menurut mereka, informasi adalah "jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima". Artinya, dengan adanya informasi tingkat kepastian menjadi meningkat. Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Abdul Kadir, 2003: 31).

II.3. Sistem Informasi

II.3.1. Pengertian Sistem Informasi

Ada beberapa definisi sistem informasi sebagaimana akan diuraikan berikut di bawah ini.

1. Menurut Alter (1992) Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

2. Menurut Bodnar dan Hopwood (1993) Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
3. Menurut Gelinas, Oram, dan Wiggins (1990) Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
4. Menurut Hall (2001) Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
5. Menurut Turban, McLean dan Wetherbe (1999) Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
6. Menurut Wilkinson (1992) Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Dari berbagai definisi tersebut dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi indormasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Abdul Kadir, 2003: 10-11).

II.3.2. Komponen Sistem Informasi

Dalam suatu sistem informasi terdapat komponen-komponen sebagai berikut:

1. Perangkat keras (*hardware*), mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instrukti yang memungkinkan perangkat keras untuk dapat memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkita keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*database*), yaitu sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyampaian data.
6. Jaringan kompiter dan komunikasi data, yaitu sistem penghubung yang memungkinkan sesumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai (Kusrini dan Andi Koniyo, 2007: 9).

II.4. Sistem Informasi Geografis

II.4.1. Konsep Dasar Geografi

Pada dekade terakhir, data spasial telah berkembang menjadi bentuk digital (dibantu oleh perangkat komputer) sehingga disebut data spasial digital (digital spatial data). Sebelum pelaksanaan SIG dilakukan dengan cara digital, SIG dilakukan dengan cara manual, yaitu dengan teknik tumpang susun dan skoring. Hasilnya

berupa uraian dan dalam bentuk peta. Selain itu, dikenal pula istilah data dasar digital (digital data base). Data dasar digital merupakan data yang dihasilkan dari pengolahan peta di dalam layar komputer(on screen maps). Data dasar digital kemudian diolah oleh fungsi-fungsi analisa pada perangkat lunak (software) yang tersedia melalui menu-menu atau petunjuk-petunjuk. Contohnya, untuk menghitung jarak, luas, atau menentukan posisi suatu fenomena menggunakan kemampuan perangkat lunak tertentu (Tatok Gunawan, dkk, 2007).

II.4.2. Pengertian Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) merupakan salah satu model sistem informasi yang banyak digunakan untuk membuat berbagai keputusan, perencanaan, dan analisis. SIG memiliki perbedaan pokok dengan sistem informasi lain. Perbedaan ini justru menjadi ciri karakteristiknya. Pada sebuah sistem informasi selain SIG, basis data atributal adalah fokus dari pekerjaan sistem, sedangkan SIG mengaitkan data atributal dengan data spesial. SIG memberi analisis keruangan terhadap data atribu tersebut. SIG menjelaskan di mana, bagaimana, dan apa yang terjadi secara keruangan yang diwujudkan dalam gambaran peta dengan berbagai penjelasan secara deskriptif, tabular, dan grafis. Dari kemampuannya tersebut, SIG memberi dua jenis model informasi, yaitu dalam bentuk spesial dan deskriptif. SIG adalah sebuah rangkaian sistem yang memanfaatkan teknologi *digital* untuk melakukan analisa spesial. Sistem ini memanfaatkan perangkat keras dan lunak dari komputer untuk melakukan pengolahan data (Eko Budiyanto, 2004: 1-2).

Penggunaan Sistem Informasi Geografis (SIG) meningkat tajam sejak tahun 1980-an. Peningkatan pemakaian sistem ini terjadi di kalagan pemerintah, militer, akademis, atau bisnis terutaama di negara-negara maju. Perkembangan teknologi digital sangat besar peranannya dalam perkembangan penggunaan SIG dalam berbagai bidang. Hal ini dikarenakan teknologi SIG banyak mendasarkan pada teknologi digital ini sebagai alat analisis (Eko Budiyanto, 2005: 2).

Hubungan anatara bentuk spesial dan deskriptif dijelaskan secara topologis. Bentuk analisis seperti ini tidak didapat dalam berbagai sistem informasi yang lain. Dalam SIG terdapat berbagai peran dari berbagai unsur, baik manusia sebagai ahli dan sekaligus operator, perangkat alat baik lunak dan keras, serta objek permasalahan (Eko Budiyanto, 2004: 2).

II.4.3. Data Sistem Informasi Geografis

Data Sistem Informasi Geografis berupa data *digital* yang berformat raster dan vektor. Vektor menyimpan data *digital* dalam bentuk rangkaian koordinat (x, y). titik disimpan sebagai sepasang angka koordinat dan poligon sebagai rangkaian koordinat yang membentuk garis tertutup. Resolusi dari data vektor tergantung jumlah titik yang membentuk garis. Raster menyatakan data grafis dalam bentuk bujursangkar yang disimpan sebagai pasangan angka menyatakan baris dan kolm dalam suatu matrik. Titik dinyatakan dalam *grid-cell*, garis dinyatakan sebagai *grid-cell* bersambungan di satu sisi, dan poligon dinyatakan sebagai gabungan *grid-cell*

yang bersambungan di semua sisi. Resolusi dari data raster ditentukan oleh ukuran *grid-cell* (Eko Budiyanto, 2002: 5).

Sumber data digital dapat berupa citra satelit atau data foto udara *digital* serta foto udara yang terdigitasi (*scanning*). Data lain dapat berupa peta dasar terdigitasi. Citra satelit yang brasal dari Landsat TM merupakan contoh data citra *digital* dengan format raster. Foto udara *digital* dan citra satelit digunakan secara saling melengkapi. Masing-masing sumber data tersebut memiliki kelebihan dan kekurangan, terutama pada kerincian dan luasan data yang dapat diperoleh. Dengan demikian, pemantaatan kedua jenis data tersebut secara saling melengkapi sangatlah menguntungkan (Eko Budiyanto, 2002: 6).

II.4.4. Metode Perolehan Data *Digital*

Ada sedikitnya lima metode perolehan data *digital* yang dikenal saat ini yaitu :

1. Digitasi peta-peta yang ada dengan menggunakan *digitizer*.
2. *Scanning* peta.
3. Produksi peta foto *digital*.
4. Masukan manual dari koordinat terkomputasi dan perhitungan.
5. Transfer dari sumber data *digital*

Ketiga metode yang pertama, yang menghasilkan data *digital* dari peta dasar, dapat dbagi dalam tiga tahap, yaitu :

1. Penyiapan
2. Digitasi atau pelarikan (*scanning*)

3. Editing data (Eko Budiyanto, 2002: 7).

II.4.5. Pengolahan Data dalam Sistem Informasi Geografis

Dalam SIG terdapat berbagai peran dari berbagai unsur, baik manusia sebagai ahli dan sekaligus operator, perangkat alat (lunak/keras) maupun objek permasalahan. SIG adalah sebuah rangkaian sistem yang memanfaatkan tenkologi *digital* untuk melakukan analisis spesial. Sistem ini memanfaatkan perangkat keras dan lunak untuk melakukan pengolahan data seperti :

1. Perolehan dan verifikasi
2. Kompilasi
3. Penyimpanan
4. Pembaruan dan perubahan
5. Manajemen dan pertukaran
6. Manipulasi
7. Penyajian
8. Analisis (Tor Berdhardsen, 1992: 23 dalam Eko Budiyanto, 2002: 3).

II.5. Basis Data (*Database*)

II.5.1. Pengertian Basis Data

Database (basis data) merupakan kumpulan data yang saling berhubungan satu dengan lainnya yang tersimpan di perangkat keras komputer dan diperlukan suatu perangkat lunak untuk memanipulasi basis data tersebut. Buku telepon, katalog *file*m merupakan contoh dari basis data (Junindar, 2008: 19).

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur *file* yang mernyusun sebuah *database* adalah Data *Record* dan *Field*.

a. Data adalah satu satuan informasi yang akan diolah, sebelum diolah data dikumpulkan di dalam suatu *file database*.
b. *Record* adalah data yang isinya merupakan satu kesatuan seperti Nama*User* dan *Password*. Setiap keterangan yang mencakup Nama*User* dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut rocord (*record number*).
c. *Field* adalah sub bagian dari *record*, dari contoh di atas maka terdiri dari 2 *field*, yatu: *Field* Nama*User* dan *Password* (Anhar, 2010: 45-46).

II.5.2. Sistem Manajemen Basis Data

Keberhasilan suatu sistem informasi sangat dipengaruhi oleh sistem basis data. Sistem basis data merupakan salah atu elemen penyusun sistem. Apabila sistem basis data ini benar-benar lengkap, akurat dan mudah untuk ditampilkan kembali maka hal itu akan meningkatkan kualitas dari sistem manajemen basis data. Untuk mengetahui perbedaan antara basis data dengan sistem basis data, kita harus

mengetahui definisi keduanya. Menurut James F. Courtney dan David B. Paradice, sistem basis data mempunyai pengertian sebagai berikut: “Sistem basis data adalah kumpulan basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer yang mendukungnya” (Sutanta, 1996 dalam Kusrini dan Andri Koniyo, 2007: 139).

Sistem manajemen basis data (*database* management sistem/DBMS) adalah perangkat lunak yang digunakan untuk mengendalikan data, termasuk penyimpanan data, pengambilan data, keamanan data, dan integrasi data. Fungsi utama DBMS adalah untuk menyediakan lingkungan yang nyaman dan efisien untuk digunakan dalam pengambilan dan penyimpanan informasi di basis data (Junindar, 2008: 19).

Pengertian basis data ini diperjelas oleh James Martin (1990), yang mengatakan sebgai berikut: “Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data denganc ara-cara tertentu sehingga mudah untuk digunakan dan ditampilkan kembali, dapat digunakan untuk satu atau lebih program aplikasi secara optimal, data dapat disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, serta disimpan sedemikian rupa sehingga penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol” (Kusrini dan Andri Koniyo, 2007: 140).

II.5.3. Arsitektur Basis Data

Terdapat dua bentuk arsitektur sistem basis data, yaitu sistem terpusat dan sistem *client-server*. Sistem basis data terpusat adalah sistem basis data yang dijalankan pada sistem komputer tunggal dan tidak berinteraksi dengan sistem pada komputer lain. Pengguna terkoneksi ke komputer pusat melalui terminal. Sedangkan sistem basis data *client-server* adalah sistem basis data yang memisahkan program pengguna dengan program basis data di sistem yang berbeda. Pengguna terkoneksi ke pusat data yang disebut *server* sistem melalui suatu program pengguna (*user interface*) yang terdapat pada komputer. Sistem tempat program pengguna berada disebut client *system* (Junindar, 2008: 20).

II.5.4. Elemen Basis Data

Adapun elemen-elemen sistem manajemen basis data adalah sebagai berikut:

a. *Database*, *Database* adalah sekumpulan dari sitem data yang saling berhubungan satu sama lain, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer, dan harus menggunakan *software* untuk melakukan manipulasi tertentu.
b. *File*, *File* adalah kumpulan *record* sejenis yang mempunyai panjang elemen dan atribut yang sama, namun valuenya berbeda. *Database* dibentuk dari kumpulan *file*. *Field* di dalam pemrosesan aplikasi dapat dikategorikan ke dalam beberapa tipe, diantaranya adalah sebagai berikut:
    1) *File* induk (master *file*)

a) *File* induk acuan (*reference master file*): *file* induk yang *record*nya relatif statis, jarang berubah nilainya. Misalnya *file* daftar gaji, *file* mata pelajaran.

b) *File* induk dinamik (*dynamic master file*): *file* induk yang nilai *record-record*nya sering berubah atau sering dimuthakirkan (*update*) sebagai hasil dari suatu transaksi. Misalnya *file* induk data barang, yang setiap saat harus di *update* bila terjadi transaksi.

2) *File* transaksi (*transaction file*), *File* ini bisa disebut *input file*, digunakan untuk merekam data hasil transaksi yang terjadi. Misalnya *file* penjualan yang berisi data hasil transaksi penjualan.

3) *File* Laporan (*report file*), *File* ini bisa disebut *output file*, yaitu *file* yang berisi informasi yang akan ditampilkan.

4) *File* sejarah (*history file*), *File* ini bisa disebut *file* arsip (*archival file*), merupakan *file* yang berisi data masa lalu yang sudah tiadk aktif lagi tetapi masih disimpan sebagai arsip.

5) *File* pelindung (*backup file*), *File* ini merupakan salinan dari *file-file* yang masih aktif di dalam *database* pada suatu saat tertentu. *File* ini digunakan sebagai pelindung atau cadangan bila *file database* yang akif mengalami kerusakan atau hilang.

c. *Record*, *Record* adalah kumpulan elemen yang saling berkaitan yang menginformasikan tentang satu enitas secara lengkap. Satu *record* mewakili satu data atau informasi.

d. *Field*, *Field* adalah bagian tertentu dari data dalam *record* yang mewakili satu entitas, misalnya *file* anggota dapat dilihat dari *field*nya, seperti kode anggota, nama dan lain-lain.

e. *Data Value*, Data *value* adalah data aktual atau informasi yang disampaikan pada setiap data elemen atau *field* data, misalnya *field* nama anggota memiliki data value Susi, Widi dan sebagainya.

f. *Enity*, *Enity* (entitas) adalah objek riil yang dapat dibedakan satu sama lain dan tidak saling bergantung. Misal, pada bidang sirkulasi, entitasnya adalah anggota dan buku.

g. *Query, Query* merupakan perintah yang dirancang untuk memanggil keloompok *record* tertentu dari satu *file* atau lebih untuk melakukan operasi pada *file*.

h. *View*, *View* adalah data yang terdiri dari sejumlah *record* yang diproses dalam urutan penampilan (Kusrini dan Andri Koniyo, 2007: 141-143).

II.5.5. Relational *Database* Management System (RDBMS)

RDBMS merupakan sekumpulan data yang disimpan sedemikian rupa sehingga mudah diambil informasinya bagi pengguna, dan data tersebut saling berhubungan. RDBMS merupakan paket perangkat lunak yang kompleks yang digunakan untuk memanipulasi *database.*

Ada tiga prinsip dalam RDBMS, yaitu :

1. Data *definition*

Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara relasi data, validasi data dan lainnya.

2. Data *manipulation*

   Data yang telah dibuat dan didefinisikan akan dikenai beberapa perlakuan, seperti penyaringan, peng-*query*-an dan lain sebagainya.

3. Data *control*

   Bagian ini berkenaan dengan cara mengendalikan data, seprti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *user*, dan sebagainya.

Semua operasi *input* dan *output* yang berhubungan dengan *database* harus menggunakan DBMS. Bila pemakai akan mengakses databae, DBMS akan menyediakan penghubung (*interface*) antara pemakai dengan *database* (Kusrini dan Andri Koniyo, 2007: 143).

II.5.6. Kamus Data

Kamus data adaah deskripsi formal mengenai seluruh elemen yang tercakup dalam DAD. Pada tahapan perancangan elemen-elemen pada kamus data akan menjadi bahan untuk menyusun basis data (Abdul Kadir: 2007, 45).

Contoh penggunaan kamus data adalah sebagai berikut:

Dosen = (NIP, Nama Dosen, Jurusan, Bidang Keahlian, Alamat)

Matakuliah = (Kode Matakuliah, nama, sks, semester, pilihan)

Jadwal = (NIP, Kode Matakuliah, hari, jam, lokal)

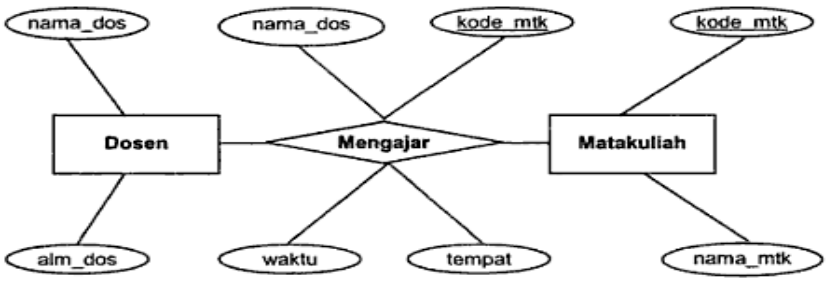
Mahasiswa = (Nopb, Nama, tempat lahir, tanggal lahir, agama, jenis kelamin, PA, alamat)

Skripsi = (Nopb, NIP, Judul Skripsi, Tanggal Mulai, Tanggal Selesai) (Yuherfizard; 2008: 33-34).

II.5.7. ERD (*Enity Relation Diagram*)

ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Enititas menunjukan hubungan antar data. Pada akhirnya ERD bis juga digunakan untuk menunjukkan aturan-aturan bisnis yang ada pada sistem informasi yang akan dibangun (Hanif Al Fatta: 2007, 121-122).

Diagram E-R digunakan untuk menggambarkan secara sistematis hubungan antar entity-entity yang ada dalam suatu sistem *database* menggunakan simbol-simbol sehingga lebih mudah dipahami. Contoh diagram E-R disajikan sebagai berikut.



Gambar II. 4: Contoh Diagram E-R
(Sumber : Yuherfizard; 2008: 17)

II.5.8. Normalisasi

Normalisasi adalah suatu proses yang bertujuan menciptakan struktur-struktur entity yang dapat mengurangi redudansi data dan meningkatkan stabilitas *database*. Ada dua fungsi normalisasi, yaitu:

a. Dapat digunakan sebagai metodologi dalam menciptakan desain *database*.
b. Dapat digunakan sebagai verifikasi terhadap hasil desain *database* yang telah dibuat, baik menggunakan E-R Model atau menggunakan model relasi, seperti yang Anda buat di atas atai dari model yang lain (Yuherfizard: 2008, 37-38).

Beberapa bentuk normalisasi *database* dapat disajikan pada Gambar berikut :

1. Bentuk Normalisasi Pertama

Bentuk normal tahap pertama terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak (multivalued attribute) atau lebh dari satu atribut dengan nilai domain yang sama.

Tabel 2. 1 Tabel Pemasok Normal I

Kd_Pemasok	NamaPemasok	C_Person	AlamatPemasok	TeleponPemasok
TK01	Toko bima	Agus	Jl.Dipanegoro 29, Gorontalo	(0435) 823944
UD01	Ud Loak Jaya	Fery	Jl. P. Hidayat 11, Gorontalo	(0435) 829834

Tabel 2. 2 Tabel Pelanggan Normal I

Kd_Pelanggan	NamaPelanggan	C_Person	AlamatPelanggan	..
P01	Toko tiga jaya	Hj.ahmad	Jl. Sudirman, 1 Gorontalo	..
P02	Ud mitra tex	Lisna	Jl. Maligu, 02 Gorontalo	..

..	TeleponPelanggan
..	(0435) 821348
..	(0435) 825515

Tabel 2. 3 Tabel Bahan Baku Normal I

Kd_BahanBaku	NamaBahanBaku	Ukuran	Satuan	HargaJualBahanBaku	..
A01	Besi Siku	2.5 x 2.5	Ujung	27900	..
A02	Besi Siku	4 x 4	Ujung	50700	..
B03	Besi Beton	4%	Ujung	10900	..

..	Stock Awal	Stock Min	Stock Max
..	10	15	50
..	20	20	55
..	5	15	30

Tabel 2. 4 Tabel Pembelian Normal I

NoNotaBeli	Tanggal	Kd_Pemasok	Kd_BahanBaku	HargaBeli	JumlahBarang	..
NB01	28/02/2007	TK01	A01	27500	10	..
NB01	28/02/2007	TK01	A02	50000	10	..
NB02	28/02/2007	UD01	B03	10500	50	..

..	DisconBeli	CaraBayar	JthTempo	UangMuka	JmlHutang
..	20000	Tunai	-	-	
..	20000	Tunai	-	-	-
..	30000	Kredit	28/03/2007	200000	600000

Tabel 2. 5 Tabel Penjualan Normal I

NoNotaJual	TglJual	TglOder	TglSelesai	Kd_Pelanggan	NamaBarangJadi	..
NJ01	21/03/2007	14/03/2007	20/03/2007	P01	Pintu Pagar	..
NJ01	22/03/2007	14/03/2007	20/03/2007	P01	Pintu Pagar	
NJ01	23/03/2007	14/03/2007	20/03/2007	P01	Pintu Pagar	
NJ02	22/03/2007	15/03/2007	21/03/2007	P02	Pintu Dorong	..
NJ02	22/03/2007	15/03/2007	21/03/2007	P02	Pintu Dorong	

..	Kd_BahanBaku	JmlBarang	UpahTenagaKerja	BiayaOverhead	UangMuka	..
..	A01	5	1000000	250000	100000	..
	A02	4	1000000	250000	100000	..
	B03	3	1000000	250000	100000	..
	A02	2	500000	350000	100000	
	B03	2	500000	350000	100000	

..	DiscJual	CaraBayar	Jumlah Biaya	JthTempo	SisaBayar	JmlPiutang
..	20000	Tunai	1625000	-	625000	-
..	20000	Tunai	1625000	-	625000	-
..	20000	Tunai	1625000	-	625000	-
	30000	Kredit	2000000	28/03/2007	1500000	1500000
	30000	Kredit	2000000	28/03/2007	1500000	1500000

Tabel 2. 6 Tabel Akun Normal I

KodeAkun	NamaAkun	SaldoAwal
111111	Kas	500000
211111	Piutang	
122222	Persediaan Bahan Baku	
311111	Pembelian	
611111	Hutang	
711111	Modal	500000

Tabel 2. 7 Tabel Jurnal Normal I

NoJurnal	Tanggal	NoBukti	Uraian	KodeAkun	..
J01	01/02/2007	A01	Pembelian Bahan Baku Tunai	311111	..
J01	01/02/2007	A01	Pembelian Bahan Baku Tunai	111111	..
J02	02/03/2007	B02	Penjualan Barang Jadi Kredit	111111	
J02	02/03/2007	B02	Penjualan Barang Jadi Kredit	211111	
J02	02/03/2007	B02	Penjualan Barang Jadi Kredit	811111	

..	Debet	Kredit
..	825000	0
..	0	825000
	500000	0
	1000000	0
	0	1500000

Tabel 2. 8 Tabel User Normal I

User_Name	Password	StatusUser
Admin	admin	Administrator
Hery	Hery	Data Entry
Nina	N1n4	Kasir
Ance	4nc5	Pimpinan

2. Bentuk Normalisasi Kedua

Bentuk normal tahap kedua terpenuhi jika normalisasi tahap pertama terpenuhi dan semua atribut tidak termasuk dalam kunci primer secara utuh. Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci fieldnya. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya. Sebuah tabel dikatakan tidak memenuhi normalisasi bentuk kedua jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key).

Gambar II. 5: Bagan Normalisasi Bentuk Ke II
(Sumber : Kusrini dan Andri Koniyo,2007: 107)

3. Bentuk Normalisasi Ketiga

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak mempunyai hubungan transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada primary key secara menyeluruh.

Gambar II. 6: Bagan Normalisasi Bentuk Ke III
(Sumber : Kusrini dan Andri Koniyo,2007: 108)

II.5.9. Perancangan Basis Data

Perancanga basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Penyusunan basis data ini

berlandaskan kamus aliran data yang telah dibuat pada tahapan sebelumnya. Perancangan basis data secara konseptual terdiri atas tiga langkah yaitu penentuan entitas pada basis data, pendefinisian hubungan antara entitas, dan penerjemahan hubungan ke dalam entitas (Abdul Kadir: 2007, 45).

II.6. Mengenal MySQL

II.6.1. Pengertian MySQL

MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database* Management System) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Porstagre SQL, dan lain-lain. MySQL merupakan DBMS yang multithread, multi *user* yang bersifat gratis dibawah lisensi GNU *General Public Licence* (GPL). Tidak seperti Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki penulisnya masing-masing (Anhar, 2010: 21).

MySQL (MyStructure Query Language) adalah salah satu *Database* Managemen Sistem (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postrahre SQL, dan lainnya. MySQL berfungsi untuk mengolah *database* menggunakan bahasa SQL. MySQL bersifat open source sehingga kita bisa menggunakannya secara gratis (Anhar, 2010: 45).

II.6.2. Konsep Dasar Pemrograman MySQL

Tempat penyimpanan data berupa informasi dalam sebuah tabel disebut dengan *database*. Program yang digunakan untuk mengolah dan mengelola *database* adalah MySQL yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur, dan menampilkan data (Anhar, 2010: 45).

II.7. Pengenalan UML (*Unified Modeling Language*)

Unified Modelling Language adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan disain berorientasi obyek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh 3 pendekar "berorientasi obyek", yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson, UML menjadi bahasa yang bisa digunakan untuk berkompilasi dalam perspektif obyek antara user dengan devloper, antara developer dengan developer, antara developer analisis dengan developer disain, dan antara developer disain dengan developer pemrograman.

UML memungkinkan developer melakukan permodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan visual membantu untuk menangkap struktur dan kelakuan dari obyek, mempermudah penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.

Namun karena UML hanya merupakan bahasa untuk pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan disain berorientasi obyek. Untuk mengetahui bagaimana melakukan analisis dan disain berorientasi obyek secara baik, sudah terdapat beberapa metodologi yang bisa diikuti, seperti Metode Booch, Metode Coad and Yourdan, Metode Jacobson, Metode Rumbaugh, Metode Wirfs-Brock, atau mungkin mengikuti metode pengembangan sistem Relational Unified Process (Julius Hermawan, 2007: 7).

Pada umumnya metode-metode yang ditujukan untuk pembangunan aplikasi berorientasi objek menggunakan UML untuk memodelkan berbagai artefak dari perangkat lunak. UML adalah sekumpulan simbol dan diagram untuk memodelkan *software*. Dengan menggunakan UML, desain *software* dapat diwujudkan dalam bentuk simbol dan diagram. Desain dalam bentuk simbol dan diagram, kemudian dapat diterjemahkan menjadi kode program. Telah tersedia tools yang dapat membuat kode program berdasar UML *Class Diagram*. Implementasi kode program dari diagram UML dapat menggunakan bahasa pemrograman apa saja dengan syarat bahasa pemrograman tersebut mendukung pemrograman berorientasi objek (Farid Aziz, 2005: 116).

II.7.1. *Actor*

Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi *actor* inji bisa berupa orang, perangkat keras, atau mungkin juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh *actor* adalah

memberikan informasi pada sistem dan/atau memerintahkan sistem untuk melakukan sesuatu.

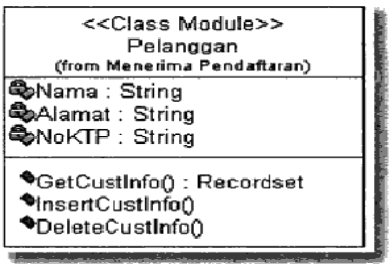
Gambar II. 7: Notasi *Actor*
Sumber : Julius Hermawan (2007: 14)

II.7.2. *Class*

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*. *Class* digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. *Class* bisa memperentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

Notasi *class* berbentuk persegi panjang berisi 3 bagian persegi paling atas untuk nama *class*, persegi panjang paling bawah untuk operasi, dan persegi panjang di tengah untuk atribut.

Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas mempresentasikan informasi yang disimpan di dalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek dan menggunakan kata kerja.



Gambar II. 8: Notasi *Class*
Sumber : Julius Hermawan (2007: 14)

II.7.3. *Interface*

Interface merupakan sekumpulan operasi tanpa implementasi dari satu *class*. Implementasi operasi dalam *interface* dijabatkan oleh operasi dalam *class*. Oleh karena itu keberadan *interface* selalu disertai *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.

Gambar II. 9: Notasi *Interface*
Sumber : Julius Hermawan (2007: 15)

II.7.4. *Use Case*

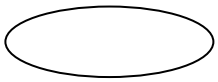
Use Case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *Use Case* hanya menjelaskan yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

Di dalam *Use Case* terdapt teks untuk menjelaskan urutan kegiatan yang disebut *Use Case specification*. *Use Case specification* terdiri dari :

1. Nama *Use Case*: mencantumkan nama dari *use case* yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan aktivitas.
2. Deskripsi singkat (*brief description*): menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari *Use Case* ini.
3. Aliran normal (*basic flow*): ini adalah jantung dari *Use Case*. Menjelaskan interaksi antara *actor* dan sistem dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar, tiada halangan atau hambatan dalam mencapau tujuan dari *Use Case*.
4. Aliran alternatif (*alternate flow*): merupakan pelengkap dari *basic flow* karena tidak ada yang sempurna dalam setiap kali *Use Case* berlangsung. Di dalam alternate *flow* ini dijelaskan apa yang akan terjadi bila suatu halangan atau hambatan terjadi sewaktu *Use Case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi, misalnya karena sistem kekurangan data untuk diolah (usia pegawai belum di*input*), terjadi masalah internal sistem komputer (folder terhapus), terjadi masalah eksternal (printer belum di *turn-on*).
5. *Special requirement*: berisi kebutuhan lain yang belum tercakup dalam aliran normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *Use Case*, sementara special requirement yang tidak berhubungan dengan kebutuhan fungsional, misalnya

kecepatan transaksi maksimum berapa cepat dan berapa lama, kapasitas akses yaitu jumlah *user* yang akan mengakses dalam waktu bersamaan.

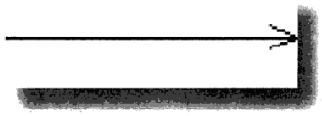
6. *Pre-condition*: menjelaskan persyaratan yang harus dipenuhi sebelum *Use Case* bisa dimulai.
7. *Post-condition*: menjelaskan kondisi yang berubah atau terjadi saat *Use Case* selesai dieksekusi.



Gambar II. 10: Notasi *Use Case*
Sumber : Julius Hermawan (2007: 16)

II.7.5. Interaction

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek. Biasanya interaction ini dilengkapi juga dengan teks, bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

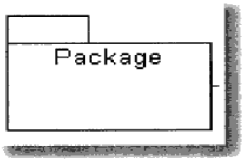


Gambar II. 11: Notasi Interaction
Sumber : Julius Hermawan (2007: 18)

II.7.6. Package

Package adlaah kontainer atau wadah konseptual yang digunakan untuk mengelompokan elemen-elemen darisistem yang sedang dibangun, sehingga bisa

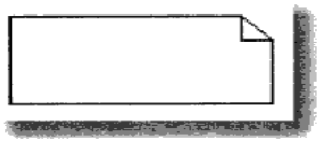
dibuat *mode* yang lebih sederhana. Tujuannya adaah untuk mempermudah pengelihatan (*visibility*) dari model yang sedang dibangun.



Gambar II. 12: Notasi Package
Sumber : Julius Hermawan (2007: 19)

II.7.7. Note

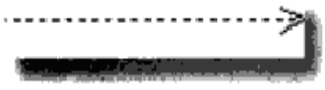
Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam *model*. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.



Gambar II. 13: Notasi Note
Sumber : Julius Hermawan (2007: 19)

II.7.8. Dependency

Dependency merupakan relasi yang menunjukkan bahwa perubahan apda salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.

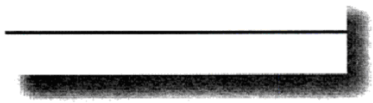


Gambar II. 14: Notasi Dependency

Sumber : Julius Hermawan (2007: 20)

II.7.9. Association

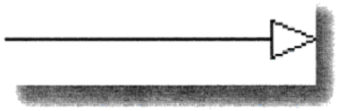
Association menggambarkan navigasi antar *class* (*navigation*), berapa banyak objek lain yang bisa berhubungan dengan satu objek (*multiplicity* antar *class*), dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).



Gambar II. 15: Note Association
Sumber : Julius Hermawan (2007: 21)

II.7.10.Generalization

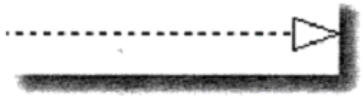
Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization*, *class* yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari *class* yang lebih umum (*superclass*), atau ”*subclass* is a *superclass*”. Dengan menggunakan notasi *generalization* ini konsep *inheritance* dari prinsip hirarki dimodelkan.



Gambar II. 16: Note Generalization
Sumber : Julius Hermawan (2007: 22)

II.7.11.Realization

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada dibagian dengan panah. Misalnya *class* merealisasikan *package*, *component* merealisasikan *class* atau *itnerface.*

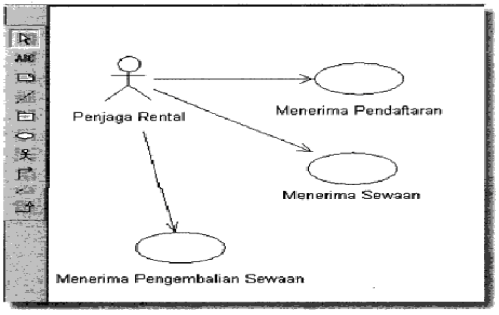


Gambar II. 17: Note Realization
Sumber : Julius Hermawan (2007: 22)

II.7.12.*Use Case* Diagram

Use Case diagram (UCD) menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. UCD menjadi dokumen kesepakatan antara *Customer*, *User* dan *Developer*. *User* menggunakan dokumen UCD ini untuk memahami sistem dan mengevaluasi bahwa benar yang dilakukan sistem adalah untuk memecahkan masalah yang *user* ajukan atau sedang dihadapi. *Developer* menggunakan dokumen UCD ini sebagai rujukan yang benar dalam mengembangkan sistem.

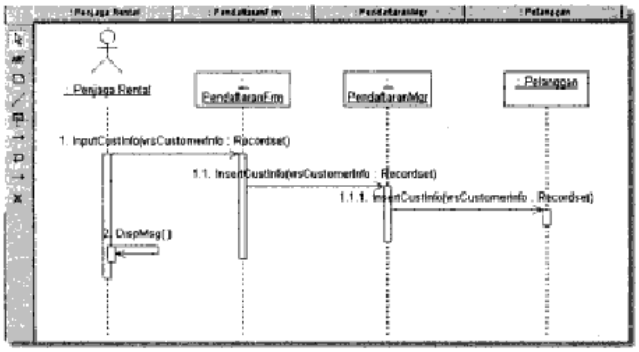
Use Case Diagram pada umumnya tersusun dari elemen *actor*, *Use Case*, *dependency*, *generalization*, dan *association*. UCD ini memberikan gambaran statis dari sistem yang sedang dibangun dan merupakan artifak dari proses analisis.



Gambar II. 18: Note *Use Case* Diagram
Sumber : Julius Hermawan (2007: 23)

II.7.13.*Sequence Diagram*

Squence diagram menjelaskan secara detil 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 intormasi yang diperlukan oleh masing-masing operasi. Pembuatan *squence* diagram merupakan aktivitas yang paling kritikal dari proses diasin karena artifak nilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program. Oleh karena itu berharga untuk meluangkan waktu lebih lama di pembuatan *sequence* diagram ini untuk menghasilkan squence diagram yang terdisain dengan baik.

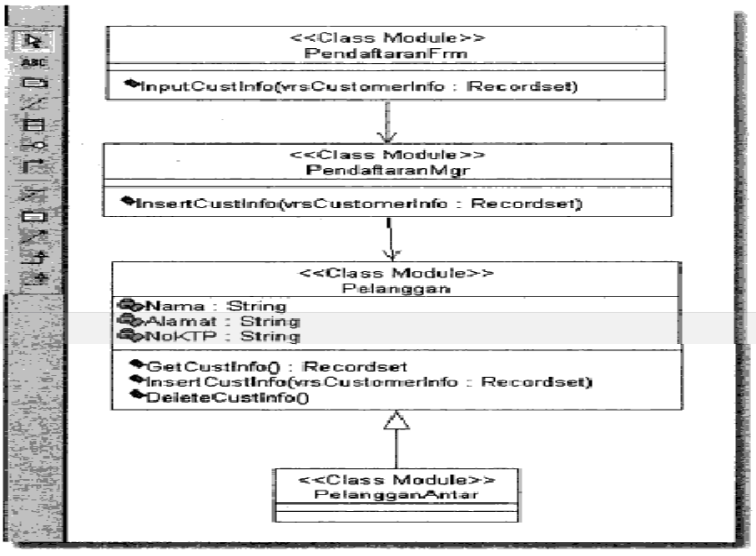


Gambar II. 19: Note Sequence Diagram
Sumber : Julius Hermawan (2007: 24)

II.7.14.*Class* Diagram

Sama seperti halnya *class*, maka *class* diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai tujuan.

Class diagram umumnya tersusun dari elemen *Class*, *Interface*, *Dependency*, *Generalization* dan *Association*. Relasi dependency menunjukkan bagaimana ketergantungan terjadi antar *class* yang ada. Relasi generalization menunjukkan bagaimana suatu *class* menjadi super*class* dari *class* lainnya dan *class* yang lain tesebut menjadi sub*class* dari *class* tersebut (Julius Hermawan, 2007: 14-28).



Gambar II. 20: Note *Class* Diagram
Sumber : Julius Hermawan (2007: 27)

II.8. Pengenalan ArcVIEW

ArcView merupakan salah satu perangkat lunak desktop Sistem Indormasi Geografis dan pemetaan yang telah dikembangkan oleh ESRI. Dengan ArcView, pengguna dapat memiliki kemampuan-kemampuan untuk melakukan visualisasi, meng-explore, menjawab query (baik basis data spasial maupun non-spasial), menganalisis data secara geografis dan sebagainya (Eddy Prahasta, 2009: 1).

ArcView mengorganisasikan sistem perangkat lunaknya sedemikian rupa sehingga dapat dikelompokkan ke dalam beberapa komponen penting sebagai berikut:

1. Project

Project merupakan suatu unit organisasi tertinggi di dalam ArcView. Project, di dalam ArcView, mirip *project* yang dimiliki oleh bahasa-bahasa pemrograman komputer (C/C++, Pascal/Delphi, Basic dan sebagainya), atau paling tidak merupakan suatu file kerja yang dapat digunakan untuk menyimpan, mengelompokkan dan mengorganisasikan semua komponen-komponen program: *view*, *theme*, *table*, *chart*, *layout*, dan *script* dalam satu kesatuan yang utuh.

2. Theme

Themes merupakan suatu bangunan dasar sistem ArcView. Themes merupakan kumpulan dari beberapa layer ArcView yang membentuk suatu tematik tertentu. Sumber data yang dapat direpresentasikan sebagai theme adalah shapfile, coverage (ArcInfo) dan citra raster.

3. View

View mengorganisasikan theme. Sebuah view merupakan representasi grafis informasi spasial dan dapat menampung beberapa *layer* atau *theme* informasi spasial (titik, garis, poligon dan citra raster). Sebagi contoh, posisi-posisi kota (titik), sungai-sungai (garis), dan batas propinsi (poligon) dapat membentuk theme dalam sebuah view.

4. Table

Sebuah tabel merupakan representasi data ArcView dalam bentuk sebuah tabel. Sebuah tabel akan berisi informasi deskriptif mengenai layer tertentu. Setiap baris data (*record*) mendefinisikan sebuah entry (misalnya ifnormasi mengenai salah satu poligon batas propinsi) di dalam basis data spasialnya. Setiap kolom (*field*) mendefinisikan atribut atau karakteristik dari entri (misalnya nama, luas, keliling, atau populasi suatu propinsi) yang bersangkutan. Dari sisi pengguna, tanpa memperhatikan sumber-sumbernya, semua tabel adalah sama. ArcView mendefinisikan templete standart untuk merujuk tabel yang diakses.

5. Chart

Chart merupakan representasi grafis dari *resume* tabel data. Chart juga merupakan hasil suatu *query* terhadap suatu tabel data. Bentuk chart yang didukung oleh ArcView adalah line, bar, column, xy scater, area, dan pie.

6. Layout

Layout digunakan unutk menggabungkan semua dokumen (*view*, tabel dan *chart*) ke dalam suatu dokumen yang siap cetak (biasanya dipersiapkan untuk pembuatan *hardcopy*).

7. Script

Script merupakan bahasa (semi) pemrograman sederhana (makro) yang diguankan untuk mengotomasikan kerja ArcView. ArcView menyediakan bahasa sederhana ini dengan sebutan Avenue. Dengan *Avenue*, pengguna dapat memodifikasi tampilan (*user interface*) ArcView, membuat program, menyederhanakan tugas-tugas yang kompleks, dan berkomunikasi dengan aplikasi-aplikasi lain (misalnya dengan ArcInfo, basis data relational atau lembar kerja elektronik). Singkatnya, dengan *script*, ArcView dapat dicustomized sedemikian rupa hingga dapat secara optimal memenuhi kebutuhan pengguna untuk tugas-tugas dan aplikasi tertentu (Eddy Prahasta, 2009: 5-7).