

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Secara umum, sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterkaitannya (inter-relasi) di dalam (usaha) mencapai suatu tujuan (atau sasaran bersama tertentu). Atau dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (subsistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan (Eddy Prahasta ; 2009 : 89).

##### **II.1.1. Konsep Dasar Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

Menurut Jerry FithGerald, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Menurut Ludwig Von Bartalanfy, sistem merupakan seperangkat unsure yang saling terkait dalam suatu antar-relasi di antara unsur-unsur tersebut dengan lingkungan.

Menurut Anatol Raporot, sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

Menurut L. Ackof, sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya (Asbon Hendra ; 2012 : 157).

### **II.1.2. Syarat - syarat Sistem**

- a. Sistem harus dibentuk untuk menyelesaikan tujuan.
- b. Elemen sistem harus mempunyai rencana yang ditetapkan.
- c. Adanya hubungan di antara elemen sistem.
- d. Unsur dasar dari proses (arus informasi, energy, dan material) lebih penting dari pada elemen sistem.
- e. Tujuan organisasi lebih penting dari pada tujuan elemen (Asbon Hendra ; 2012 : 157).

### **II.1.3. Karakteristik Sistem**

#### **a. Komponen (Component)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem, tidak peduli betapa pun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut *supra sistem*. Sebagai contoh suatu perusahaan dapat disebut dengan suatu sistem dan industri

yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem,. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

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

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

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

Environmen merupakan segala sesuatu di luar batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnakan atau dikendalikan agar tidak mengganggu operasi sistem.

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

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

**e. Masukan Sistem (*Input*)**

Merupakan energy yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (***Maintenance Input***) adalah energy yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (***Signal Input***) adalah energy yang diproses untuk didapatkan keluaran. Sebagai contoh, di dalam sistem computer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

**f. Keluaran Sistem (*Output*)**

Merupakan hasil dari energi yang diolah oleh sistem, meliputi output yang berguna, contohnya informasi yang dikeluarkan oleh computer. Dan output yang tidak berguna dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

**g. Pengolah Sistem (*Process*)**

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada computer, bagian produksi yang mengubah bahan baku menjadi barang jadi, serta bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

**h. Tujuan Sistem (*Goal*)**

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu

mengenai sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya (Asbon Hendra ; 2012 : 156-162).

## II.2. Informasi

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi informasi = *input* – proses – *output*.

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relative, tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya.

Kualitas informasi tergantung dari tiga hal, yaitu informasi harus :

- a. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan juga harus jelas mencerminkan maksudnya.
- b. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.
- c. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda (Asbon Hendra ; 2012 : 167).

### **II.3. Sistem Informasi**

Suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya. Sebuah sistem terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi. Sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan.

Sekelompok orang, prosedur, input, output, dan pengolahannya secara bersama-sama menghasilkan informasi yang akurat, tepat waktu, dan relevan bagi penggunanya.

Menurut Robert A. Leitch, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Ada empat operasi dasar dari sistem informasi, yaitu mengumpulkan, mengolah, menyimpan, dan menyebarkan, informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau ke luar organisasi (Asbon Hendra ; 2012 : 158).

### **II.4. Sistem Informasi Geografis**

Pada dasarnya pengertian sistem informasi geografis merupakan gabungan dari tiga unsur pokok : sistem, informasi, geografis. Dengan demikian, pengertian

terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG. Dengan melihat unsur-nsur pokoknya, maka sudah jelas bahwa SIG juga merupakan salah satu tipe sistem informasi seperti yang telah dibahas diatas. Tetapi dengan tambahan unsur “Geografis”, atau SIG merupakan suatu sistem yang menekankan pada unsur “Informasi Geografis”.

Dengan memperhatikan pengertian sistem informasi diatas, maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek penting yang terdapat di permukaan bumi. Jadi SIG juga merupakan sejenis perangkat lunak, perangkat keras (manusia, prosedur, basis data, dan fasilitas jaringan komunikasi) yang dapat digunakan untuk memfasilitasi proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran, data/informasi geografis berikut atribut-atribut terkait. (Eddy Prahasta : 2009 ; 109).

#### **II.4.1. Subsistem SIG**

SIG dapat diuraikan menjadi beberapa sub-sistem yaitu sebagai berikut :

- a. Data Input** : Sub-sistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya dari berbagai sumber. Sub-sistem ini pula yang bertanggung jawab dalam mengonversikan atau mentransformasikan format-format data aslinya ke dalam format (native) yang dapat digunakan oleh perangkat SIG yang bersangkutan.
- b. Data Output** : Sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengeksponnya ke format yang dikehendaki) seluruh atau

sebagai basis data (spasial) baik dalam bentuk softcopy maupun hardcopy seperti halnya table, grafik, report, peta, dan lain sebagainya.

- c. Data Management** : sub-sistem ini mengorganisasikan baik data spasial maupun table-tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau di-retrieve (di-load ke memori), di-update, dan di-edit.
- d. Data Manipulation & Analysis** : Sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, sub-sistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsi-fungsi dan operator matematis & logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan (Eddy Prahasta : 2009 ; 118).

#### II.4.2. Komponen / Arsitektur SIG

SIG merupakan salah satu sistem yang kompleks dan pada umumnya juga (selain yang stand-alone) terintegrasi dengan lingkungan sistem computer lainnya di tingkat fungsional dan jaringan (network). Jika diuraikan, SIG sebagai sistem terdiri dari beberapa komponen (sebagai berikut) dengan berbagai karakteristiknya (Gistut 94).

- a. Perangkat Keras.** Pada saat ini SIG sudah tersedia bagi berbagai *platform* perangkat keras ; mulai dari kelas PC *desktop*, *workstations*, hingga *multi-user* host yang bahkan dapat digunakan oleh banyak orang secara bersamaan (simultan) dalam jaringan computer yang luas, tersebar, berkemampuan tinggi, memiliki ruang penyimpanan (*harddisk*) yang besar, dan mempunyai kapasitas

memori (RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat pada karakteristik-karakteristik fisik perangkat kerasnya sehingga keterbatasan memori pada suatu PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi SIG adalah computer (PC), *mouse*, monitor (plus VGA-card grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver GPS*, dan *scanner*.

**b. Perangkat Lunak.** Dari sudut pandang yang lain, SIG bisa juga merupakan sistem perangkat lunak yang tersusun secara modular dimana sistem basis datanya memegang peranan kunci. Pada kasus perangkat SIG tertentu, setiap sub-sistem di implementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (\*.exe) yang masing-masing dapat dieksekusi tersendiri.

**c. Data & informasi geografi.** SIG dapat mengumpulkan dan menyimpan data atau informasi yang diperlukan baik secara tidak langsung (dengan cara meng-*import*-nya dari format-format perangkat lunak SIG yang lain) maupun secara langsung dengan cara melakukan digitasi data spasialnya atau manual dengan menggunakan *digitizer*) dari peta analog dan kemudian memasukkan data atributnya dari table-tabel atau laporan dengan menggunakan *keyboard*.

**d. Manajemen.** Suatu proyek SIG akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang-orang memiliki keahlian (kesesuaian dengan job-description yang bersangkutan) yang tepat pada semua tingkatan (Eddy Prahasta : 2009 ; 120).

## II.5. ArcView

ArcView adalah salah satu diantara sekian banyak perangkat lunak yang digunakan dalam sistem informasi Geografis. ArcView memiliki kemampuan yang tinggi dalam pembuatan peta digital hingga analisis spasial (Eko Budiyanto ; 2010).

Pada tahun 1991, ESRI mengembangkan perangkat lunak ArcView untuk digunakan di komputer *desktop*. Arcview memiliki tampilan yang lebih menarik, interaktif, memiliki tingkat kemudahan yang tinggi hingga lebih terkenal dan sering digunakan dewasa ini. Hampir semua pengguna aplikasi SIG telah mengenal ArcView. Demikian banyaknya pengguna ArcView, hingga ESRI lebih giat lagi dalam mengembangkannya. Pada saat ini ArcView telah dikembangkan lebih lanjut hingga mencapai bidang-bidang yang sebelumnya tidak terbayangkan. Pengembangan ArcView lanjut ini banyak disediakan dalam bentuk modul-modul tambahan atau *extension* untuk kebutuhan-kebutuhan aplikasi-aplikasi khusus (Eddy Prahasta : 2009 : 9)

## II.6. UML (*Unified Modeling Language*)

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membandu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

Bahasa pemodelan grafis telah ada di industri perangkat lunak sejak lama. Pemicu umum dibalik semuanya adalah bahwa bahasa pemrograman berada pada

tingkat abstraksi yang tidak terlalu tinggi untuk memfasilitasi diskusi tentang desain.

UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem yang berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997, UML menghancurkan menara Babel tersebut menjadi sejarah. (Prabowo Pudjo Widodo : 2011 : 6-9)

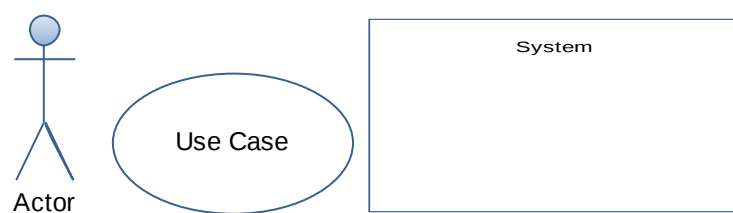
### **II.6.1. Diagram-Diagram UML**

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

#### **1. Use Case Diagram**

Use-case adalah konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna potensial. Use-case terdiri dari sekumpulan skenario

yang dilakukan oleh seorang aktor (orang, perangkat keras, urutan waktu atau sistem yang lain). Sedangkan use-case diagram memfasilitasi komunikasi di antara analis dan pengguna serta diantara analis dan klien. Diagram use case menunjukkan 3 aspek dari sistem yaitu : actor, use-case, dan system boundary. Actor adalah pengguna sistem, biasa nya mewakili peran orang, sistem yang lain atau alat yang berkomunikasi dengan use-case. Use Case adalah tugas yg dilakukan oleh actor. Sekumpulan use-case biasa nya dikelompokkan dalam suatu group yang disebut System Boundary. (Radiant Victor Imbar dan Yuliusman Kurniawan : 2012 : 4. Simbol usecase ditunjukkan pada gambar II.5.



**Gambar II.1. Use Case Diagram**

Sumber : (Prabowo Pudjo Widodo dan Herlawati : 2010 : 20).

## 2. Activity Diagram

Activity diagrams 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 (Radiant Victor Imbar dan Yuliusman Kurniawan : 2012 : 5).

| Simbol                                                                            | Keterangan                          |
|-----------------------------------------------------------------------------------|-------------------------------------|
|  | Titik Awal                          |
|  | Titik Akhir                         |
|  | Activity                            |
|  | Pilihan untuk pengambilan keputusan |

**Gambar II.2. Simbol-simbol pada Activity Diagram**

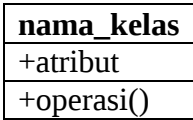
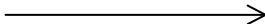
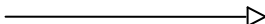
Sumber : (Radiant Victor Imbar dan Yuliusman Kurniawan : 2012 : 5).

### 3. Class Diagram

Class diagram membantu dalam visualisas istruktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang palingbanyak. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain(dalam logical view) dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap decían, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Class diagram juga merupakan pondasi untuk component diagram dan deployment diagram. (Prastuti Sulistyorini : 2009 : 4).

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class

diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. (Haviluddin : 2011 : 3).

| Simbol                                                                                                                                | Deskripsi                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Kelas<br>                                            | Kelas Pada struktur sistem                                                                                                            |
| Antarmuka / <i>interface</i><br><br>nama_interface   | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek                                                              |
| Asosiasi / <i>assosiation</i><br>                    | Relasi antar kelas dengan maksna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>                                     |
| Asosiasi berarah / <i>directed assosiation</i><br> | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dangan <i>multiplicity</i> |
| generalisasi<br>                                   | Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus)                                                             |
| Keberuntungan / <i>dependency</i><br>              | Relasi antar kelas dengan makna kebergantungan antar kelas                                                                            |
| Agregasi / <i>agregation</i>                                                                                                          | Relasi antar kelas dengan makna                                                                                                       |

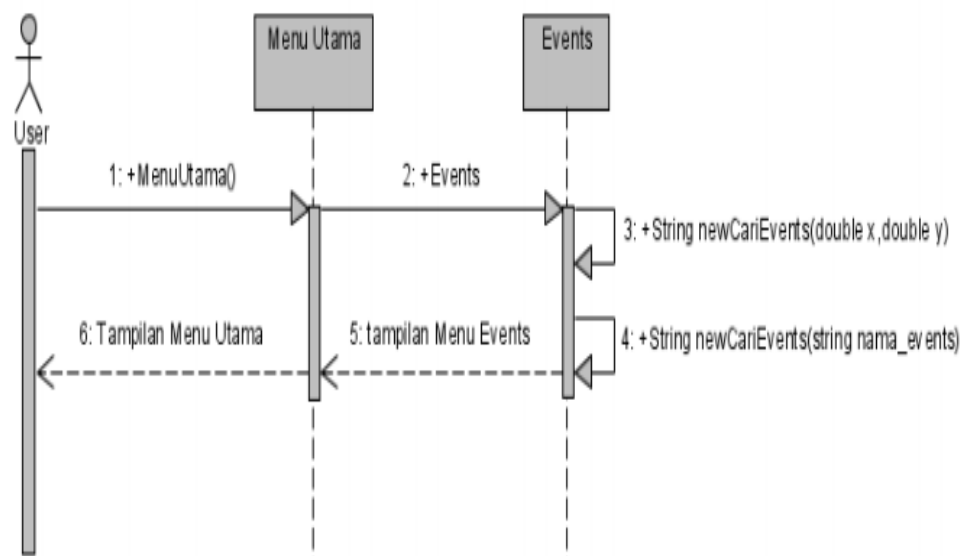
**Gambar II.3. Contoh Simbol Class Diagram**

Sumber : (Rosa A.S dan M. Shalahuddin : 2011 : 123)

#### 4. Squence diagram

Diagram sequence menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan usecase. Sequence diagram memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case. Diagram sequence sebaiknya digunakan diawal tahap desain atau analisis karenakesederhanaannya dan mudah untuk dimengerti. Sequence diagram menjelaskan interaksi objek yang disusun

berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi(urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan usecase diagram. (Haviluddin : 2011 : 5)



**Gambar II.4. Contoh Squence Diagram**  
 Sumber : (S. Nofan Maulana Rachman : 2012 : 9)

## II.7. Basis Data (*Database*)

Basis data merupakan istilah yang sangat penting dan paling sering digunakan di dalam dunia sistem informasi geografis. Demikian pentingnya basis data hingga SIG-pun tidak dapat dipisahkan darinya. SIG memerlukan basis data dan SIG memiliki bagian yang berkemampuan untuk mengelola basis datanya secara *inherent*. Oleh karena itu, SIG-pun merupakan sistem pengelola basis data (Eddy Prahasta ; 2009 :315).

### II.7.1. Database 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* (tidak perlu bayar untuk menggunakannya) pada berbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial).

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

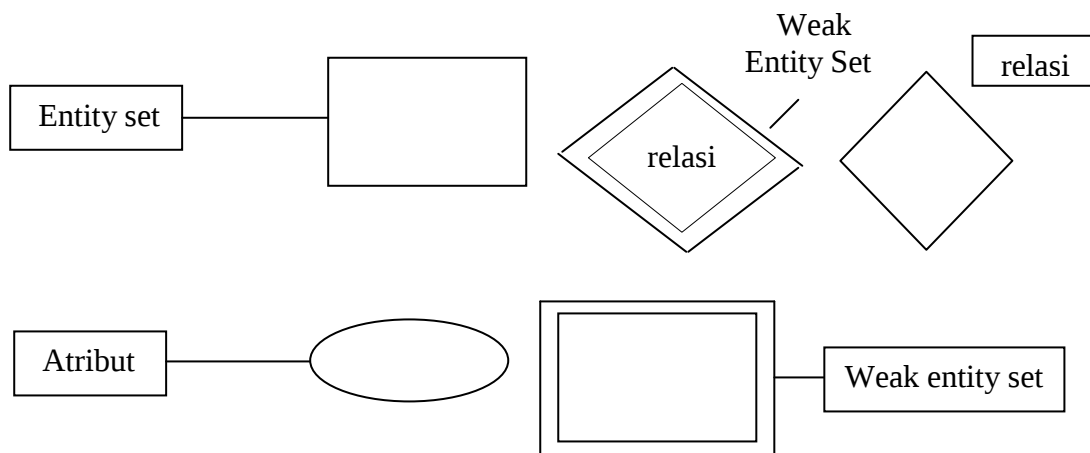
### II.8. ERD (*Entity Relationship Diagram*)

Model *entity-relationship* (ER) pertama kali diperkenalkan oleh Charles Bachman pada tahun 1969-an. Kemudian, diagram yang juga mendeskripsikan struktur data ini di populerkan oleh Pin-Shan Peter Chen pada tahun 1976.

Menurut Eddy Prahasta (2009: 172) Diagram ERD yang telah berisi komponen-komponen *entity set* (representasi tabel) dan *relationship set* (yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari sebagian dunia nyata) dapat digambarkan lebih baik dan sistematis. Adapun symbol-simbol dan notasi yang digunakan di dalam penulisan diagram ini adalah :

1. Persegi panjang dengan garis tunggalnya mempresentasikan *entity set* (biasa, normal, atau kuat), sementara yang menggunakan garis ganda merupakan *weak entity set*.
2. *Ellips* yang menyatakan atribut-atribut *entity set*.
3. Belah ketupat (*diamond*) dengan garis tunggalnya menggambarkan *relationship set* antara dua *entity set* biasa, sementara yang menggunakan garis ganda mewakili relasi yang terjalin antara *entity set* biasa dengan *weak entity set*.
4. Garis yang menghubungkan antara *entity set* (tabel) dengan atribut-atributnya dan antara *entity set* dengan *relationship set*-nya.

Berikut ini merupakan gambar tampilan Elemen Dasar Diagram ER :

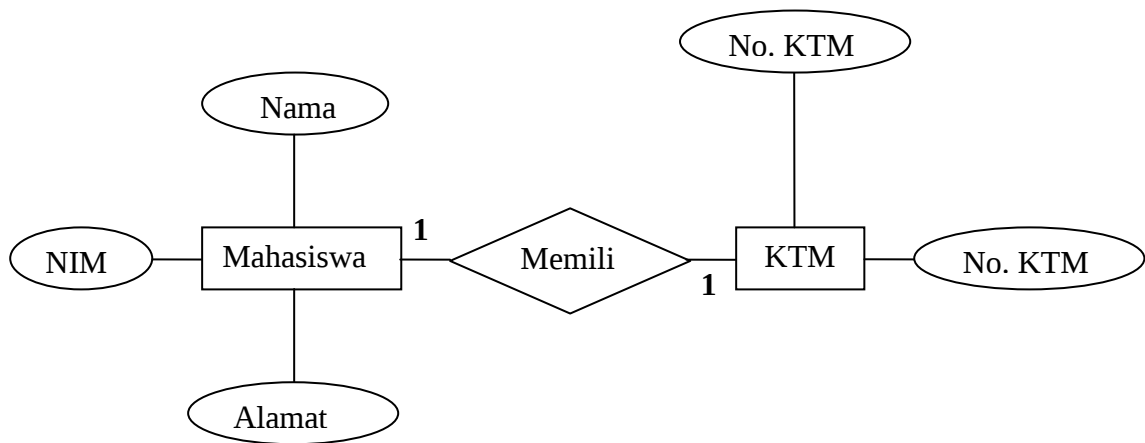


**Gambar II.5 : Contoh Tampilan Elemen Dasar Diagram ER**  
(Sumber : Eddy Prahasta ; 2009 : 173)

### II.8.1. Diagram ER Untuk Relasi Satu Kesatu

Sebagai ilustrasi, diagram ER berikut memperlihatkan adanya relasi antara *entity set* (tabel) Mahasiswa dengan *entity set* (tabel) KTM. *Relationship set*-nya

dinamai “Memiliki”. Dengan relasi ini, setiap mahasiswa akan memiliki satu KTM setiap saatnya. Dan setiap KTM akan dimiliki oleh seorang mahasiswa.



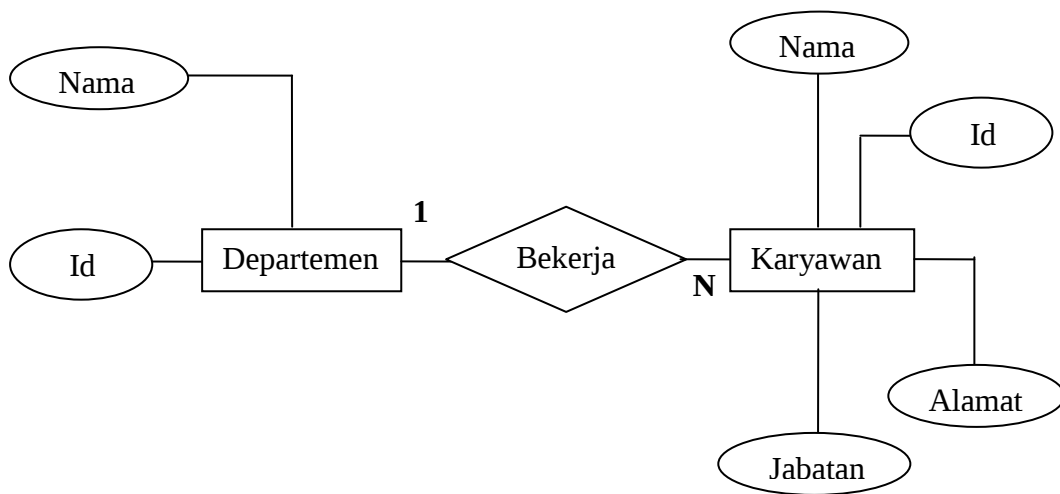
**Gambar II.6 : Contoh Tampilan Diagram ER dengan Relasi Satu ke Satu**  
(Sumber : Eddy Prahasta ; 2009 : 173)

### II.8.2. Diagram ER Untuk Relasi Satu Kebanyak

Sebagai ilustrasi, diagram ER berikut memperlihatkan adanya relasi antara *entity set* departemen. Pada contoh ini, *relationship set*-nya dinamai “Bekerja”.

Dengan relasi ini, setiap departemen terdapat banyak karyawan yang bekerja di dalamnya dan setiap karyawan hanya dapat bekerja dalam departemen saja.

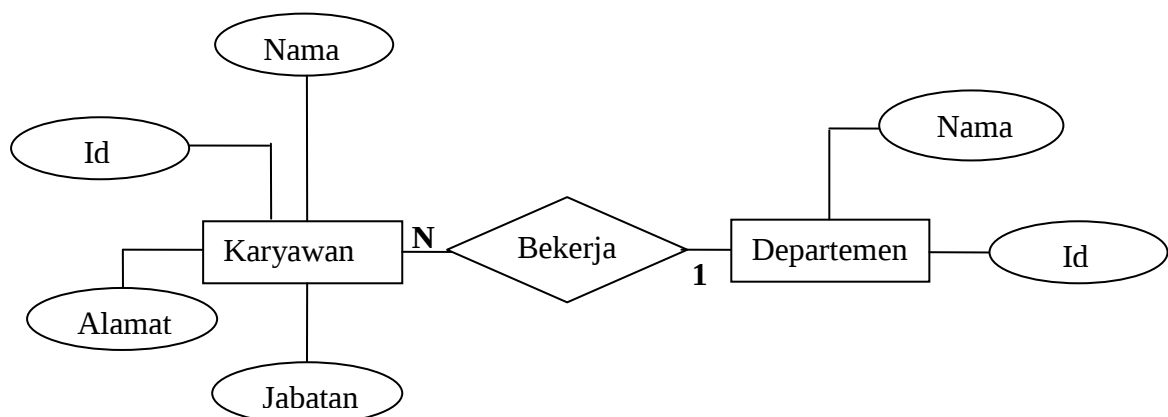
Berikut ini merupakan gambar dari ilustrasi di atas :



**Gambar II.7 : Contoh Tampilan Diagram ER dengan Relasi Satu ke Banyak**  
(Sumber : Eddy Prahasta ; 2009 : 175)

### II.8.3. Diagram ER Untuk Relasi Banyak Kesatu

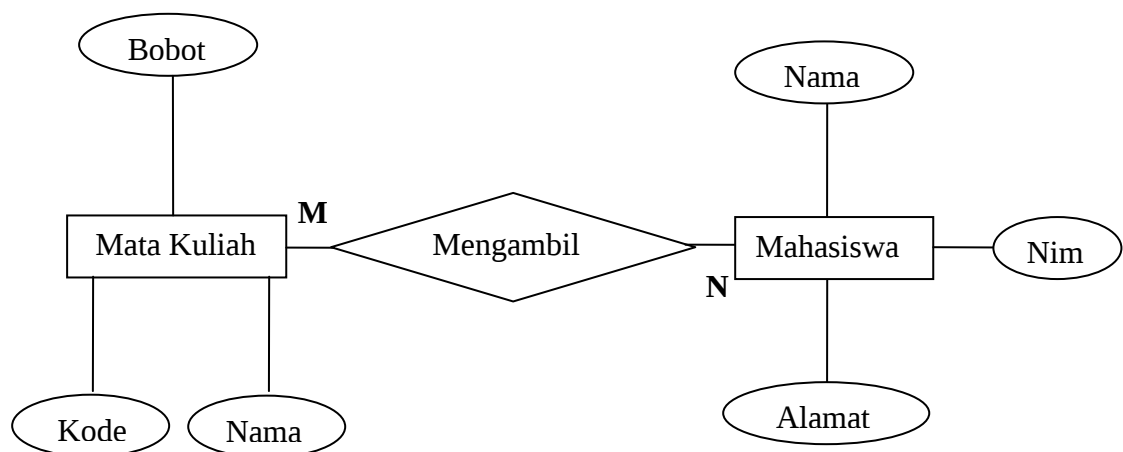
Sebagai ilustrasi, diagram ER berikut memperlihatkan adanya relasi antara *entity set* karyawan dengan *entity set* departemen. Pada contoh ini, *relationship set*-nya dinamai sebagai “Bekerja”. Dengan relasi ini, banyak karyawan bekerja di satu departemen atau setiap karyawan hanya dapat bekerja dalam satu departemen. Berikut ini merupakan gambar dari ilustrasi di atas :



**Gambar II.8 : Contoh Tampilan Diagram ER dengan Relasi Banyak Ke Satu**  
(Sumber : Eddy Prahasta ; 2009 : 175)

#### II.8.4. Diagram ER Untuk Relasi Banyak Ke Banyak

Sebagai ilustrasi, diagram ER berikut memperlihatkan adanya relasi antara *entity set* mahasiswa dengan *entity set* mata kuliah. *Relationship set*-nya dinamai “Mengambil” (atau “mempelajari”). Dengan relasi ini, setiap mahasiswa (banyak) dapat mengambil lebih dari satu (banyak) mata kuliah. Begitu pula dengan setiap mata kuliah yang dapat diambil oleh lebih dari satu (banyak) mahasiswa. Berikut ini merupakan gambar dari ilustrasi di atas :



**Gambar II.9 : Tampilan Diagram ER dengan Relasi Banyak Ke Banyak**  
(Sumber : Eddy Prahasta ; 2009 : 176)

#### II.9. Normalisasi

Setelah melalui tahapan di atas atau ERD, maka hasil pada diagram tersebut mulai direlasasikan pada tabel-tabel database. Untuk itu diperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses di mana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan

terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci. Misalnya pada sebuah tabel data\_siswa ada field nomor induk data field nama siswa serta field tanggal lahir. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut:  $nmr\_induk \rightarrow nm\_siswa$  dan  $nmr\_induk \rightarrow tgl\_lahir$ . Artinya  $nm\_siswa$  memiliki ketergantungan fungsional terhadap  $nmr\_induk$ . Field  $nm\_siswa$  isinya juga ditentukan oleh field  $nmr\_induk$ . Maksud dari semua itu adalah  $nmr\_induk$  adalah field kunci yang menentukan karena tidak ada nomor induk yang sama pada satu sekolah, jadi field  $nmr\_induk$  dapat dijadikan patokan untuk mengisi  $nm\_siswa$  dan field lainnya.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenal dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di –dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya
2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada

pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.

3. Normal Form pertama(1<sup>st</sup> Normal Form), pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
4. Normal Form kedua(2<sup>nd</sup> Normal Form), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut .
5. Normal Form ketiga(3<sup>rd</sup> Normal Form), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal(normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga (Wahana Komputer : 2010 : 32).

## **II.10. PHP (*PHP Hypertext Processor*)**

Menurut dokumen resmi PHP, PHP merupakan singkatan dari *PHP Hypertext Processor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, bisa menampilkan isi *database* ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Pada saat ini PHP cukup populer sebagai peranti pemrograman web, terutama dilingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada server-server yang berbasis UNIX, Windows, dan Macintosh. Pada awalnya, PHP dirancang untuk diintegrasikan dengan web server Apache. Namun, belakangan PHP juga dapat bekerja dengan web server seperti PWS (Personal Web Server) dan IIS (Internet Information Server). (Abdul Kadir ; 2008 : 2).

## **II.11. MapServer**

Mapserver merupakan aplikasi freeware dan opensource yang digunakan untuk mengembangkan aplikasi-aplikasi internet-based yang melibatkan spasial. Mapserver memiliki cukup fungsionalitas inti SIG yang dapat mendukung berbagai aplikasi web terkait spasial. Selain itu, mapserver juga sangat unggul di dalam mengubah data spasial (citra, data, vektor, dan peta digital) untuk aplikasi web. Untuk menjalankan dan menampilkan peta yang dihasilkan oleh mapserver, diperlukan dua file yaitu mapfile dan html file. Mapfile berisikan konfigurasi penyajian peta yang ditulis dalam bahasa dan sintaks tersendiri. Informasi ini

kemudian diolah dan disajikan oleh program mapserver. Sedangkan file HTML digunakan untuk melakukan format penyajian hasil peta. File HTML bisa berupa HTML biasa atau template yang disisipkan sintaks mapserver atau file HTML yang disisipkan PHP Mapscript (Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI) Volume 1, Nomor 3, Desember 2012).