

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Pendukung Keputusan**

Menurut Keen dan Scoot Morton Sistem Pendukung Keputusan merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah - masalah semi struktur .

Sistem pendukung keputusan atau *Decision Support System (DSS)* adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semi terstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Konsep DSS dikemukakan pertama kali oleh Scoot Morton pada tahun 1971 (Jurnal Pelita Informatika Budi Dharma ; Nanda Abdurrahman Wahid ; 2014 :93).

##### **II.1.1. Karakteristik Sistem Pendukung Keputusan**

Karakteristik sistem pendukung keputusan yaitu :

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
- b. Adanya interface manusia/ mesin dimana manusia (user) tetap memegang kontrol proses pengambilan keputusan.

- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
- e. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

### **II.1.2. Ciri-Ciri Sistem Pendukung Keputusan**

Kriteria atau ciri-ciri sistem pendukung keputusan adalah sebagai berikut :

- a. Banyak pilihan/alternatif.
- b. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
- c. Banyak input/variabel.
- d. Ada faktor resiko, dibutuhkan kecepatan, ketepatan, dan keakuratan (Konsep Data Mining vs Sistem Pendukung Keputusan ; Dicky Nofriansyah ; 2014 : 2).

### **II.1.3. Fase Dalam Sistem Pendukung Keputusan**

Tiga fase dalam pengambilan keputusan yaitu :

- a. *Intelligence*

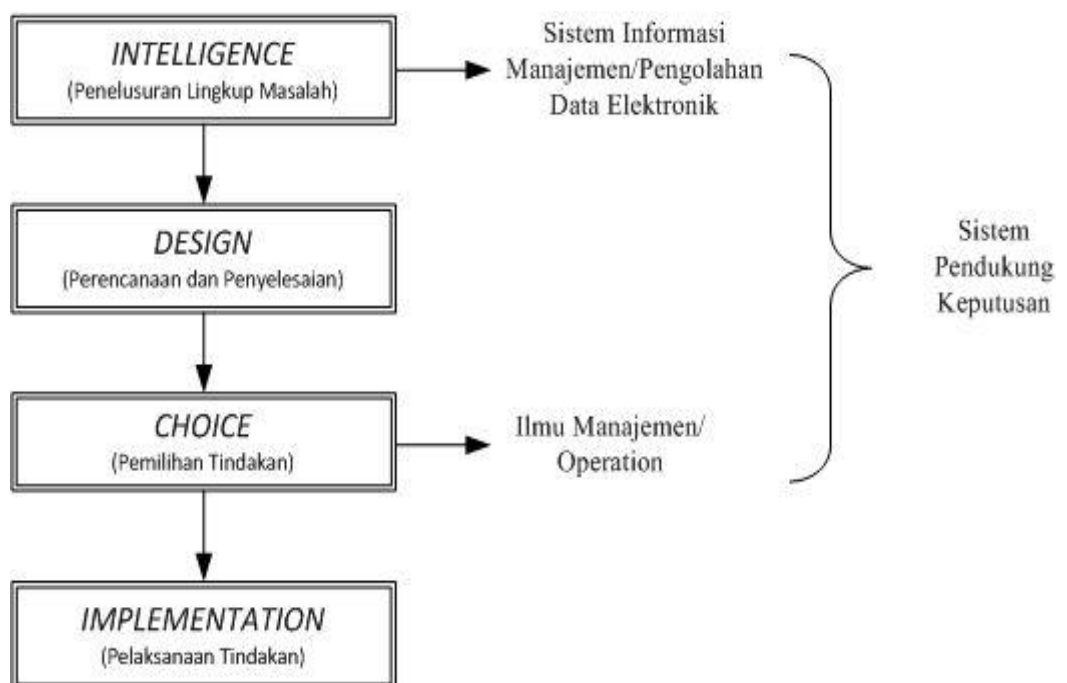
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

b. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap melakukan pengujian kelayakan solusi.

c. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan kedalam proses pengambilan keputusan (Konsep Data Mining vs Sistem Pendukung Keputusan ; Dicky Nofriansyah; 2014 : 2-3).



**Gambar II.1. Fase Proses Pengambilan Keputusan**

(Sumber : Dicky Nofriansyah ; 2014 : 2-3)

#### II.1.4. Komponen Sistem Pendukung Keputusan

Secara garis besar sistem pendukung keputusan dibangun oleh tiga komponen utama yaitu :

a. Sub sistem (*Database*)

Sub sistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan sistem manajemen basis data (*database management system*).

b. Sub sistem model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus dijaga fleksibilitasnya. Hal lain yang harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

c. Sub sistem dialog (*User Sistem Interface*)

sub sistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan sub sistem dialog. Melalui sub sistem dialog sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat

(Konsep Data Mining vs Sistem Pendukung Keputusan ; Dicky Nofriansyah ; 2014 : 3-4).

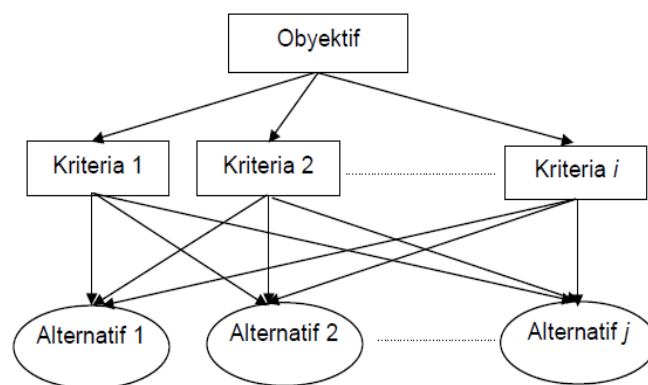
## **II.2. Metode Fuzzy MCDM (Multiple Criteria Decision Making)**

Fuzzy MCDM (*Multiple Criteria Decision Making*) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. (Kusumadewi, 2007 : 65).

Berdasarkan tujuannya, MCDM dapat dibagi menjadi 2 model : *Multi Attribute Decision Making* (MADM); dan *Multi Objective Decision Making* (MODM). Seringkali MCDM dan MADM digunakan untuk menerangkan kelas atau kategori yang sama. MADM digunakan untuk menyelesaikan masalah-masalah dalam ruang diskret. Oleh karena itu, pada MADM biasanya digunakan untuk melakukan penilaian atau seleksi terhadap beberapa alternatif dalam jumlah yang terbatas. Sedangkan MODM digunakan untuk menyelesaikan masalah-masalah pada ruang kontinyu (seperti permasalahan pada pemrograman matematis). (Zimmermann, 1991 : 23)

Dalam problem *multi criteria decision making* (MCDM), pengambil keputusan menilai sekumpulan alternatif keputusan berdasarkan sekumpulan kriteria. Salah satu pendekatan yang banyak digunakan untuk menyelesaikan persoalan MCDM ini adalah *Analytic Hierarchy Process* (Saaty, 1980).

Pada garis besarnya, ada tiga langkah yaitu (1) Dengan memiliki sejumlah kriteria  $i = 1, \dots, m$ , tentukan bobot relatif,  $w_i$  terhadap obyektif utama; (2) untuk setiap kriteria  $i$ , bandingkan alternative  $j=1, \dots, n$  dan tentukan bobot relatif  $w_{ij}$  terhadap kriteria  $i$ ; dan (3) tentukan bobot akhir  $W_j$  terhadap keseluruhan kriteria dengan menggunakan  $W_j = w_1jw_1 + w_2jw_2 + \dots + w_mjw_m$ . Selanjutnya alternatif diurutkan mulai dari alternatif yang memiliki bobot akhir terbesar.



Dalam melakukan *pairwise comparisons*, pengambil keputusan dibantu oleh skala yang terlihat pada Tabel 1 (Saaty, 1980).

**Tabel 1. Skala untuk Pairwise Comparison**

<i>Numerical rating</i>	<i>Judgment or Preference</i>	<i>Remarks</i>
1	<i>Equally important</i>	<i>Two attributes contribute equally to the attribute at the higher decision level</i>
3	<i>Moderately more important</i>	<i>Experience and judgment slightly favor one attribute over another</i>
5	<i>Strongly more important</i>	<i>Experience and judgment strongly favor one attribute over another</i>
7	<i>Very strongly more important</i>	<i>Experience and judgment very strongly favor one attribute over another; its dominance has been demonstrated in practice</i>
9	<i>Extremely more important</i>	<i>Experience and judgment extremely favor one attribute over another; the evidence favoring one attribute over another is of the highest possible order of affirmation.</i>
2,4,6,8	<i>Intermediate values</i>	<i>Used when compromise is needed</i>

Setelah keseluruhan proses *pairwise comparisons* dilakukan, matriks *pairwise comparison* berikut akan diperoleh:

$$P = \begin{bmatrix} 1 & a_{12} & \dots & a_{1n} \\ a_{21} & 1 & \dots & a_{2n} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ a_{n1} & a_{n2} & \dots & 1 \end{bmatrix} \dots\dots\dots(1)$$

dimana  $a_{ji} = a_{ij}$  dan  $n$  menunjukkan jumlah *item* yang dibandingkan. Jika terdapat  $n$  *item* yang dibandingkan satu sama lainnya, hanya diperlukan sejumlah  $n(n-1)/2$  penilaian. Dari matriks  $P$  seperti dalam persamaan, prioritas lokal,  $W$ , yang menunjukkan bobot variabel keputusan dalam level yang lebih rendah relatif terhadap satu variabel keputusan yang berada di level yang lebih tinggi diperoleh dengan persamaan *Eigen* berikut:

$$P[W] = \lambda_{\max}[W] \dots\dots\dots(2)$$

where  $\lambda_{\max}$  adalah nilai *Eigen* terbesar dari  $P$ . Ketika *pairwise comparison* yang dilakukan konsisten dengan sempurna, nilai  $\lambda_{\max}$  akan sama dengan jumlah baris dalam matriks  $P$ . Selanjutnya prioritas global, atau ranking dari alternatif dihitung melalui persamaan berikut:

$$S_i = \sum_{j=1}^N a_{ij} W_j \quad \text{for } i = 1, 2, 3, \dots, M \dots\dots\dots(3)$$

dimana:

$S_i$  = nilai *weighted score* dari alternatif  $i$

$a_{ij}$  = evaluasi alternatif  $i$  terhadap kriteria  $j$

$W_j$  = bobot kriteria  $j$

$M$  = jumlah alternatif

$N$  = jumlah kriteria

### II.2.1. Pairwise comparison dengan Fuzzy numbers

Seperti dibahas sebelumnya, tulisan ini menggunakan *fuzzy numbers* (bilangan *fuzzy*) untuk membantu pengambil keputusan menghadapi ketidakpresisian. Bilangan *fuzzy* dapat dituliskan sebagai berikut:

$$F \{ (x, (x)), x R \} A = \mu \in \dots \dots \dots (4)$$

dimana  $x$  merupakan bilangan real,  $R : -\infty < x < +\infty$  dan  $(x) A \mu$  merupakan tingkat keyakinan (*degree of belief*) dari  $x$ , yang bernilai dalam interval  $[0,1]$ . Tulisan ini menggunakan *triangular fuzzy numbers*, yang merupakan bagian dari L-R *fuzzy sets* (Dubois, 1980).

Sedangkan prioritas global, yang menunjukkan ranking dari masing-masing metoda peramalan, diperoleh dengan cara mengalikan bobot setiap kriteria  $j$  dengan nilai evaluasi dari metoda peramalan yang bersangkutan. Persamaannya dapat dituliskan sebagai berikut:

$$\tilde{P}_i = (\tilde{w}_1 \otimes \tilde{v}_{i1}) \oplus (\tilde{w}_2 \otimes \tilde{v}_{i2}) \oplus \dots \oplus (\tilde{w}_j \otimes \tilde{v}_{ij}) \dots \dots \dots (5)$$

Di sini  $\tilde{v}_{ij}$  adalah prioritas lokal untuk alternatif  $i$  relatif terhadap kriteria  $j$ . Nilai defusifikasi atau eksak dari ranking metoda peramalan diperoleh melalui *defuzzifying* terhadap prioritas global. Untuk *triangular fuzzy number*,  $(, , ) \sim (UP_i, MP_i, LP_i)$ , nilai defusifikasinya dapat diperoleh dari persamaan berikut (Tang et al 2000):

$$DP_i = \frac{[(UP_i - LP_i) + (MP_i - LP_i)]}{3} + LP_i \quad \forall i \dots \dots \dots (6)$$

Nilai defusifikasi untuk setiap metoda peramalan kemudian dinormalkan dengan membagi nilai defusifikasi tersebut dengan nilai penjumlahan semua nilai defusifikasi.

Hasil yang *fuzzy* di atas dapat dijadikan angka defusifikasi dengan menggunakan persamaan (8) dan hasilnya adalah  $S1 = 0.49$ ;  $S2 = 0.27$ ;  $S3 = 1.19$ . Atau setelah dinormalkan akhirnya diperoleh penilaian masing-masing pemasok sebagai berikut:  $(S1)N = 0.25$ ;  $(S2)N = 0.14$ ;  $(S3)N = 0.61$ . Dengan demikian konsep produk yang sedang dinilai memiliki kemungkinan “sangat mungkin” sebesar 25%, “cukup mungkin” sebesar 14% dan “gagal” 61%. Dengan cara yang sama, probabilitas dari konsep produk lainnya dapat diperoleh.

### **II.3. Kerajinan Tangan**

Kerajinan tangan adalah menciptakan suatu produk atau barang yang dilakukan oleh tangan dan memiliki fungsi pakai atau keindahan sehingga memiliki nilai jual. Kerajinan tangan yang memiliki kualitas tinggi tentu harganya akan mahal, jika kalian memiliki keterampilan dan berusaha untuk membuat suatu produk mungkin dengan kerajinan yang akan anda miliki bisa menjadi suatu usaha yang menjanjikan.

Menurut Kadjim (2011:10) kerajinan tangan adalah suatu usaha yang dilakukan secara terus meneerus dengan penuh semangat ketekunan, kecekatan, kegigihan, berdedikasi tinggi dan berdaya maju yang luas dalam melakukan suatu karya.

Kerajinan Tangan memiliki dua fungsi yaitu Fungsi Pakai dan Fungsi Hias.

1. Fungsi Pakai adalah Kerajinan yang hanya mengutamakan kegunaan dari benda kerajinan tersebut dan memiliki keindahan sebagai tambahan agar menjadi menarik.
2. Fungsi Hias adalah Kerajinan yang hanya mengutamakan keindahan tanpa memperhatikan guna dari barang tersebut, contoh kerajinan ini seperti miniatur, patung dll yang hanya menjadi kenikmatan bagi siapa yang melihatnya.

Faktor yang mempengaruhi ciri khas kerajinan suatu daerah :

1. Budaya.
2. Letak Geografis.
3. Sumber Daya Alam.

Tahap atau cara pembuatan kerajinan tangan :

1. Membuat rancangan atau desain.
2. Menyiapkan alat dan bahan.
3. Membuat benda sesuai rancangan.
4. *Finishing* ( tahap akhir ) (Tugas Akhir ; Regi Wahyu Briyansari ; 2012 : 24-25 ).

Kerajinan Tangan jika dilihat dari segi bahan dibagi menjadi 2 yaitu :

1. Bahan Organik

Limbah organik adalah jenis limbah yang berasal dari tumbuhan maupun hewan yang tergolong limbah yang mudah terurai melalui proses alami,

contoh limbah organik adalah sisa sayuran, buah-buahan, kulit dan daun. limbah organik cukup banyak dilingkungan kita, banyak orang yang sudah memanfaatkan limbah organik ini sebagai produk kerajinan. ini dikarenakan semakin banyak orang yang perhatian terhadap pemanfaatan limbah organik sebagai produk kerajinan. limbah organik yang digunakan untuk kerajinan tangan yaitu cangkang kerang laut, sabut kelapa, tempurung kelapa, bubuk teh, kulit jagung, kulit telur.

## 2. Bahan Anorganik

Limbah anorganik adalah jenis limbah yang berwujud padat, sangat sulit atau bahkan tidak bisa untuk di uraikan atau bisa membusuk, limbah anorganik tidak mengandung unsur karbon, contoh limbah anorganik adalah plastik, beling dan baja. limbah anorganik umumnya berasal dari kegiatan industri, pertambangan dan domestik yaitu dari sampah rumah tangga, contoh limbah anorganik yang digunakan untuk kerajinan tangan yaitu kaleng bekas, botol, plastik, karet sintetis, potongan atau pelat dari logam, berbagai jenis batu-batuan, pecah-pecahan gelas, tulang-belulang.

## **II.4. Basis Data**

Secara umum untuk menjelaskan tentang pengertian basis data dapat ditinjau dari dua sisi, secara kharfiah dan istilah. Menurut pengertian secara kharfiah, basis data berasal dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai suatu markas atau gudang, tempat bersarang atau tempat berkumpul. Data dapat diartikan sebagai representasi dari fakta dunia yang

mewakili suatu objek (manusia, barang, peristiwa, keadaan, dsb) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Adapun pengertian menurut istilah terdapat beberapa defenisi sebagai berikut :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa perulangan (*redudancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan tertentu.
4. Kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.

Menurut Elmasari, penggunaan istilah basis data lebih dibatasi pada arti implisit yang khusus mempunyai beberapa pengertian yaitu :

1. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real word* atau *miniworld*). Misalnya basis data perbankan, perpustakaan, pertanahan, perpajakan.
2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga apabila data terkumpul secara acak tanpa mempunyai arti, tidak dapat disebut basis data.
3. Basis data perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan tertentu.

4. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai (Basis Data : Abdul Munif ; 2013 : 7-8).

#### II.4.1. Komponen Basis Data

Basis data merupakan suatu sistem yang dibangun oleh beberapa komponen diantaranya ada enam komponen pokok antara lain :

- a. Perangkat keras (*Hardware*) dalam sistem komputer. Dalam sistem pengolahan basis data digital perangkat utama sebagai pengolah data adalah komputer.
- b. Perangkat lunak aplikasi (*Software*) lain yang mendukung dan bersifat opsional. Perangkat lunak digunakan untuk mendukung proses pengelolaan basis data. Misalnya, bahasa pemrograman c, *basic* pascal.
- c. Sistem operasi (*Operating System*) merupakan perangkat lunak yang digunakan untuk mengelola aplikasi basis data dan penggunaan sumber daya komputer.
- d. Basis data lain yang mempunyai keterkaitan dan hubungan dengan basis data itu sendiri. Berisi atau memiliki objek-objek basis data seperti, *file*, tabel, indeks. Mempunyai struktur baik untuk basis data maupun objek-objek secara detail.
- e. Sistem Pengolahan Basis Data atau *Database Management System* (DBMS) merupakan program aplikasi untuk pengolahan basis data, seperti *microsoft access*, *oracle*, dan lain-lain.

- f. Pemakai (*User*) yaitu pengguna yang terlibat dalam pengolahan basis dan penggunaan basis data (Basis Data ; Abdul Munif ; 2013 : 8).

#### **II.4.2. Tujuan Penggunaan Basis Data**

Beberapa tujuan penggunaan basis data adalah sebagai berikut :

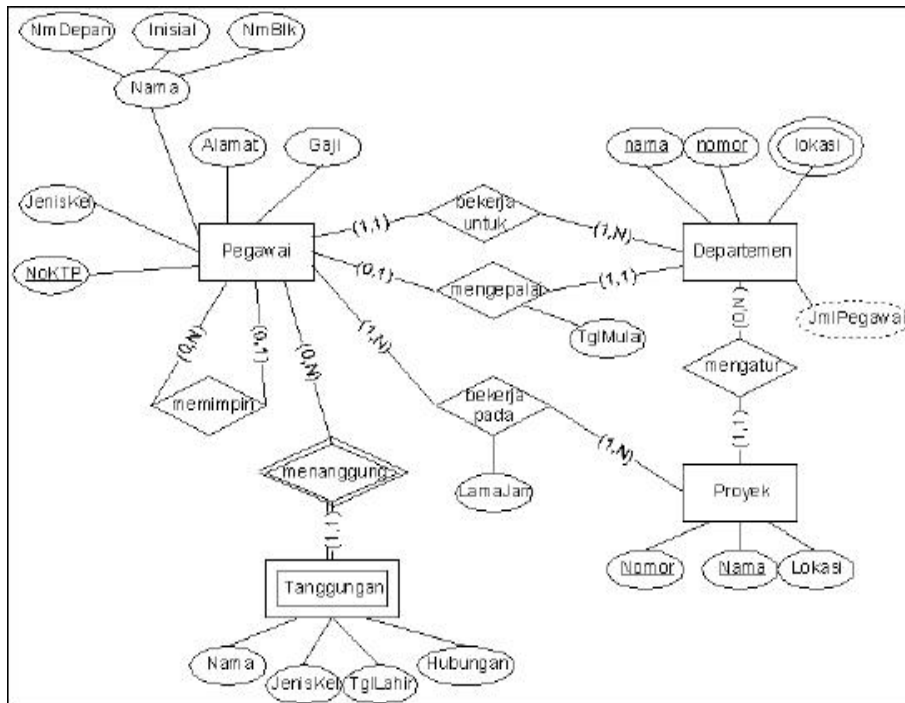
- a. Kecepatan dan kemudahan (*Speed*), melalui basis data diharapkan pengguna dapat melakukan penyimpanan, perubahan, dan menampilkan kembali dengan cepat dan mudah.
- b. Efisiensi ruang penyimpanan (*Space*), penggunaan basis data mampu mengurangi pengulangan atau redundansi data. Hal ini dapat dilakukan dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk *file*) antara kelompok data yang saling berhubungan.
- c. Keakuratan (*Accuracy*), melalui basis data keakuratan data lebih terjaga dengan menerapkan aturan dan batasan tertentu (*Constraint*), tipe data, domain data, dan keunikan data.
- d. Ketersediaan (*Availability*), dengan basis data, data yang sudah tidak dipakai dapat dipisahkan dari sistem *database* yang sedang aktif. Hal ini dapat dilakukan dengan cara penghapusan atau memindahkannya ke media *backup*, untuk menghemat ruang penyimpanan. Selain itu dapat memanfaatkan teknologi jaringan komputer agar data yang berada di suatu lokasi atau cabang dapat juga diakses oleh lokasi atau cabang lainnya.

- e. Kelengkapan (*Completeness*), agar data yang dikelola senantiasa lengkap baik relatif terhadap kebutuhan pemakai maupun terhadap waktu. Hal ini dapat dilakukan melalui penambahan *record-record* data, perubahan struktur basis data, menambah *field* pada tabel, atau menambah tabel baru.
- f. Keamanan (*Security*), walaupun tidak semua sistem basis data menerapkannya, keamanan dalam penggunaan basis data diperlakukan pada sistem yang besar dan serius. Dengan penerapan ini, setiap pengguna dibedakan hak aksesnya yakni, ditentukan objek-objek mana saja yang bisa diakses dan proses apa saja yang bisa dilakukan.
- g. Kebersamaan (*Shareability*), agar data yang dikelola oleh sistem mendukung lingkungan *multiuser* (banyak pemakai) dengan menjaga atau menghindari munculnya *problem* baru seperti inkonsistensi data (karena terjadi perubahan data yang dilakukan oleh beberapa *user* dalam waktu yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data) (Basis Data ; Abdul Munif ; 2013 : 10).

## **II.5. Entity Relationship Diagram (ERD)**

*Entity Relationship Diagram* (ERD) adalah suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam data dengan atributnya. *Entity Relationship Diagram* (ERD)

merupakan model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak (Basis Data ; Abdul Munif ; 2013 : 42).



**Gambar II.2. Contoh ERD Sistem Basis Data Kepegawaian**

(Sumber : Abdul Munif ; 2013 : 48)

### II.5.1. Relasi

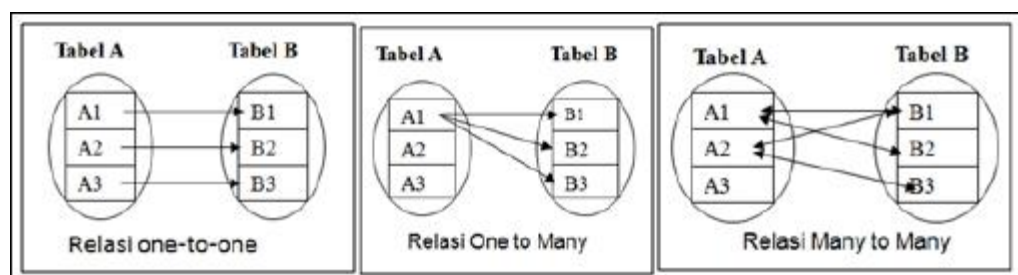
Relasi menyatakan hubungan antara dua atau beberapa entitas. Setiap relasi mempunyai batasan (*constraint*) terhadap kemungkinan kombinasi entitas yang berpartisipasi. Batasan tersebut ditentukan dari situasi yang diwakili relasi tersebut. Ragam atau jenis relasi dibedakan menjadi beberapa macam antara lain adalah :

- a. Relasi Binary merupakan relasi antara dua entitas. Relasi binary ini dibedakan menjadi :

- Relasi *One-to-one* (notasi 1:1)

- Relasi *One-to-many* (notasi 1:N) atau *many-to-one* (notasi N:1)
- Relasi *Many-to-many* (notasi M:N)

b. Relasi Ternary merupakan relasi antara tiga entitas atau lebih. Dalam relasi *One-to-one* (1:1) setiap *atribute* dari satu entitas berpasangan dengan satu *atribute* dari entitas yang direlasikan. Dalam relasi *One-to-many* (1:N) atau *many-to-one* (N:1) satu *atribute* berelasi dengan beberapa *atribute* dari entitas yang direlasikan. Dalam *Many-to-many* (M:N) satu *atribute* berelasi dengan beberapa *atribute* dari entitas yang direlasikan. Begitu pula sebaliknya (Basis Data ; Abdul Munif ; 2013 : 43).



**Gambar II.3. Ragam Relasi Antar Entitas**

(Sumber : Abdul Munif ; 2013 : 43)

## II.6. Kamus Data (*Data Dictionary*)

Menurut Jogiyanto (2005c:725) kamus data adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Kamus data digunakan pada tahap analisis sistem maupun pada tahap perancangan sistem. Pada tahap analisis kamus data dapat digunakan sebagai alat komunikasi antara analisis

sistem dengan pemakai sistem tentang informasi yang dibutuhkan oleh pemakai sistem, sedangkan pada tahap perancangan sistem kamus data digunakan untuk merancang input laporan-laporan dan database. Isi dari kamus data adalah sebagai berikut :

1. Nama arus data

Kamus data dibuat berdasarkan berdasarkan arus data yang mengalir di *data flow diagram*, maka nama arus data juga harus dicatat di kamus data sehingga dalam membaca diagram alir data memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu pada diagram alir data dapat langsung mencarinya dengan mudah di kamus data.

2. Alias

Alias atau nama lain dari data yang harus ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen yang satu dengan yang lainnya.

3. Bentuk data

Diketahui bahwa arus data dapat mengalir dari kesatuan luar suatu proses, data yang mengalir ini biasanya dalam bentuk laporan serta dokumen hasil cetakan. Bentuk dari data yang mengalir dapat berupa :

- a. Dokumen dasar atau formulir
- b. Dokumen hasil cetakan komputer
- c. Laporan tercetak
- d. Tampilan layar monitor
- e. Variabel

f. Parameter

g. Field

4. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu dicatat di kamus data agar memudahkan mencari arus data pada diagram alir.

5. Struktur data

Struktur data menunjukkan arus data yang dicatat pada kamus data yang terdiri dari elemen-elemen data.

6. *Volume*

*Volume* yang dicatat adalah tentang *volume* rata-rata yang menunjukkan banyaknya data yang mengalir dalam suatu periode tertentu, dan *volume* puncak yang menunjukkan *volume* terbanyak. *Volume* ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas dan jumlah *input*, alat proses dan alat *output*.

7. Periode

Periode ini menunjukkan kapan terjadinya arus data. Periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan *input* data dapat dimasukkan, kapan proses program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

## 8. Penjelasan

Untuk lebih memperjelas lagi tentang makna dan arus data yang dicatat kamus data, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut (Tugas Akhir ; Dea Putri Lestari ; 2013 : 33-36).

### II.6.1. Notasi Kamus Data

Notasi dalam kamus data ada dua macam yaitu :

#### a. Notasi tipe data

Notasi tipe data ini digunakan untuk membuat spesifikasi format *input* maupun *output* struktur data.

NOTASI	KETERANGAN
X	Setiap karakter
9	Angka <i>Numeric</i>
A	Karakter <i>Alphabet</i>
Z	Angka nol ditampilkan sebagai spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	<i>Hypen</i> , sebagai tanda penghubung
/	<i>Slash</i> , sebagai tanda pembagi

**Gambar II.4. Notasi Tipe Data**

**(Sumber : Dea Putri Lestari ; 2013 : 36)**

#### b. Notasi struktur data

Notasi struktur data digunakan untuk membuat spesifikasi elemen data.

Dimana notasi yang umum digunakan adalah sebagai berikut :

Notasi	Keterangan
=	Terdiri dari
+	And (dan)
()	Pilihan (Ya atau Tidak)
{}	Iterasi atau pengulangan proses
[]	Pilih salah satu pilihan
I	Pemisah pilihan didalam tanda [ ]
*	Keterangan atau catatan
@	Petunjuk ( <i>key field</i> )

**Gambar II.5. Notasi Struktur Data**  
(Sumber : Dea Putri Lestari ; 2013 : 37)

## II.7. Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisikan atau memecah data menggunakan cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan inefisiensi pengolahan. Proses normalisasi akan menghasilkan relasi yang optimal yaitu :

1. Memiliki struktur *record* yang mudah untuk dimengerti.
  2. Memiliki struktur *record* yang sederhana dalam pemeliharaan.
  3. Memiliki struktur *record* yang mudah untuk ditampilkan kembali untuk memenuhi kebutuhan pemakai.
  4. Minimalisasi kerangkapan data guna meningkatkan kinerja system
- (Basis data ; Abdul Munif ; 2013 : 89).

### II.7.1. Bentuk-Bentuk Normalisasi

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Hasil dari proses normalisasi adalah tabel–table data dalam bentuk normal (*normal form*), yaitu tabel–tabel data yang terhindar dari dua hal yaitu:

- a. Pengulangan informasi.
- b. Potensi *inkonsistensi* data pada operasi perubahan.

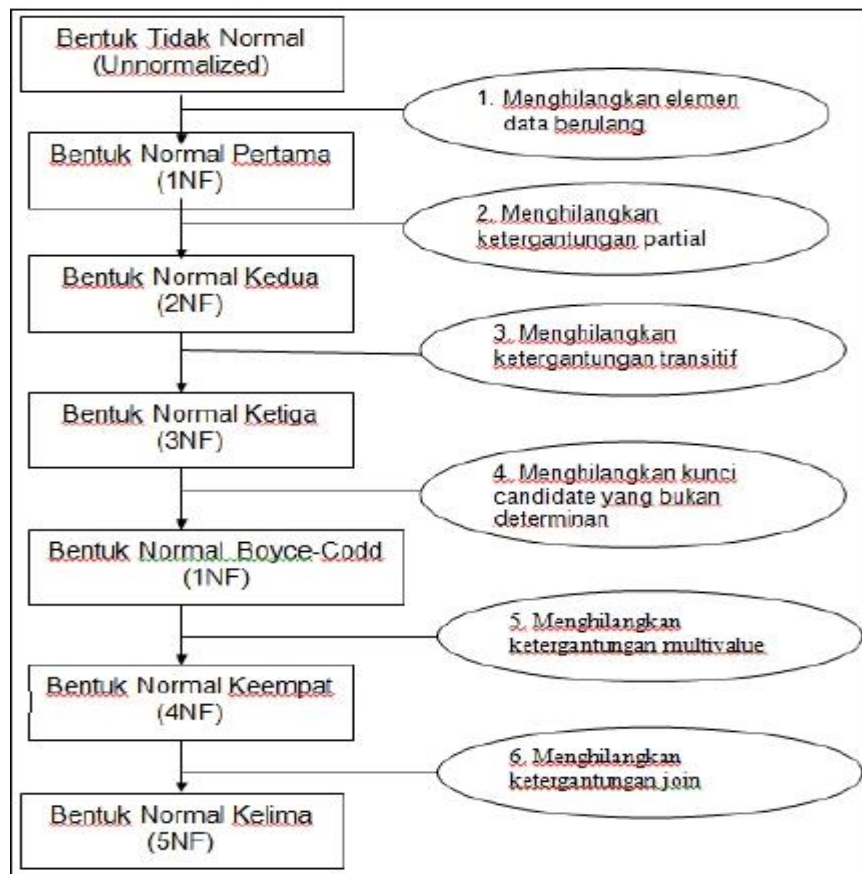
Terdapat enam bentuk normal (normal form) dalam teknik normalisasi data, keenam bentuk tersebut adalah :

- a. Bentuk Normal Tahap pertama (1st Normal Form)
- b. Bentuk Normal Tahap Kedua (2nd Normal Form)
- c. Bentuk Normal Tahap Ketiga (3rd Normal Form)
- d. Bentuk Normal Boyce - Code (BCNF)
- e. Bentuk Normal Tahap Keempat (4rd Normal Form)
- f. Bentuk Normal Tahap Kelima (4rd Normal Form) (Basis Data ; Abdul Munif ; 2013 : 103)

### II.7.2. Proses Normalisasi Data

Dalam proses normalisasi, data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat. Apabila table yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk

yang optimal. Langkah-langkah yang dilakukan dalam melakukan normalisasi data diperlihatkan dalam gambar dibawah ini (Basis Data : Abdul Munif ; 2013 : 103-104):



**Gambar II.6. Proses Normalisasi Data**  
(Sumber : Abdul Munif ; 2013 : 103-104)

a. Bentuk tidak normal (*Unnormalized Form*)

Bentuk ini memiliki ciri-ciri, yaitu :

- 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.

b. Bentuk Normal Tahap pertama (*1st Normal Form*)

Bentuk normal ke satu 1 NF ini mempunyai beberapa ciri antara lain yaitu:

- Setiap data dibentuk dalam *flat file* (file data/ rata)
- Data dibentuk dalam satu *record* demi satu *record* dan nilai dari *field-field* berupa "*atomic value*", tidak dapat dibagi-bagi lagi.
- Tidak ada set *atribute* yang berulang-ulang atau *atribute* bernilai ganda (*multivalued*).
- Tidak ada set atribut *composite* atau kombinasinya dalam *domain* data yang sama.
- Tiap field hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukanlah pecahan kata sehingga artinya lain.

nis	namasiswa	hobi1	hobi2	hobi3
111	Tiara Hadira	Memasak	Nonton	shopping
112	Hadi Saputra	Programming komputer	game online	Memancing
113	Rima Aninda	Membaca buku	bulu tangkis	
114	Nindia Sari	Membuat kue	membaca novel	golf
115	Saputra Sinara	Memancing	travelling	sepakbola
116	Diana Riani Tiara	Membaca buku	memasak	bowling
117	Tora Hadira Putra	Sepakbola	tennis	renang

**Gambar II.7. Contoh data yang belum memenuhi 1NF**  
(Sumber : Abdul Munif ; 2013 : 105)

Untuk dapat memenuhi aturan 1NF, maka dilakukan penataan ulang data (dekomposisi) menjadi 2 entitas, yakni entitas siswa dan entitas hobi seperti gambar berikut :

Tabel siswa		Tabel hobi	
nis	namasiswa	nis	hobi
111	Tiara Hadira	111	Shopping
112	Hadi Saputra	115	Sepakbola
113	Rima Aninda	117	Sepakbola
114	Nindia Sari	112	Programming
115	Saputra Sinara	111	Nonton
116	Diana Riani Tiara	114	Membuat kue
117	Tora Hadira Putra	114	Membaca novel
		113	Membaca buku
		116	Membaca buku
		116	memasak
		111	Memasak
		115	Memancing
		112	Memancing
		114	Golf
		112	Game online
		113	Bulu tangkis
		115	Travelling
		117	tennis
		117	renang

**Gambar II.8. Contoh data yang memenuhi 1NF**  
(Sumber : Abdul Munif ; 2013 : 106)

c. Bentuk Normal Tahap Kedua (*2nd Normal Form*)

Bentuk normal kedua mempunyai syarat yaitu:

- Bentuk data telah memenuhi kriteria bentuk normal kesatu.
- *Attribute* bukan kunci haruslah bergantung secara fungsi pada kunci utama atau *primary key*.
- Sudah ditentukan kunci kunci field, dimana kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

d. Bentuk Normal Tahap Ketiga (*3rd Normal Form*)

Untuk menjadi bentuk normal ketiga (3 NF) suatu tabel harus

mempunyai ciri-ciri sebagai berikut:

- Memenuhi bentuk 2 NF (normal kedua).
- Atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci utama atau *primary key*.
- Setiap attribute bukan kunci haruslah bergantung hanya pada primary key dan pada primary key secara menyeluruh (Basis Data ; Abdul Munif ; 2013 : 104-107).

## 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) pada pelbagai 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*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.(Abdul Kadir, 2008 : 348).

## II.9. Bahasa Pemrograman *PHP*

Menurut dokumen resmi PHP, PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan

dalam server dan diproses di server. Hasilnyalah 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, anda bisa menampilkan isi database kehalaman 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 (Abdul Kadir, 2008 : 2).

## **II.10. XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan

yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya.

### **II.11. *Adobe Dreamweaver CS5***

*Adobe Dreamweaver CS5* adalah versi terbaru dari Dreamweaver yang merupakan bagian dari Adobe Creative Suite 5. Dreamweaver sendiri merupakan aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga bias dikenal dengan istilah WYSIWYG (*What You See Is What You Get*), yang intinya adalah anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah site dan dapat melihat hasil desainnya secara langsung.

Dengan Kemampuan fasilitas yang optimal dalam jendela Design akan memberikan kemudahan untuk mendesain web meskipun untuk para web desainer pemula sekalipun. Kemampuan Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, JavaScript, dan yang lainnya juga memberikan fasilitas maksimal kepada desainer web dengan menyertakan bahasa pemrograman didalamnya. (Andi Offset, Yogyakarta, 2011 : 2).

### **II.12. *Unified Modelling Language (UML)***

*Unified Modelling Language (UML)* adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi

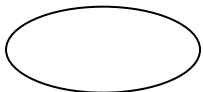
sintak dalam memodelkan sistem secara visual (Braun, *et. al.* 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, *et. al.* 2004).

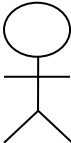


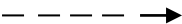
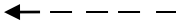
Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program (Jurnal Informatika Mulawarman ; Haviluddin ; 2011 : 1).

### II.12.1. Use Case Diagram

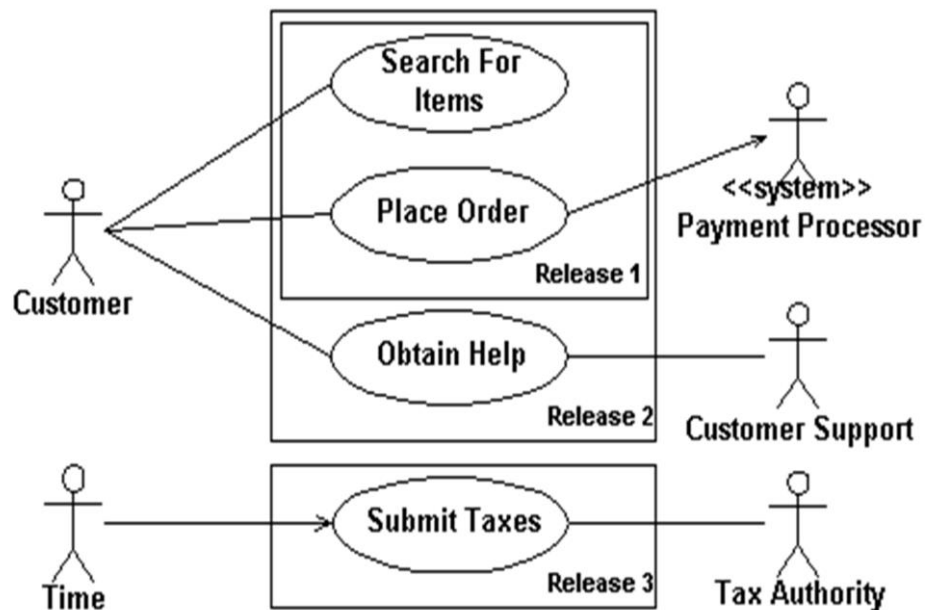
Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai *elips horizontal* dalam suatu diagram UML *use case*. (Jurnal Informatika Mulawarman ; Haviluddin ; 2011 : 4). Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

**Tabel II.4. Simbol Use Case**

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>use case</i>.</p>

	<p>Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau system bisa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan didalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Haviluddin ; 2011 : 4)



**Gambar II.9. Contoh Use Case Diagram**

(Sumber : Haviluddin ; 2011 : 4)

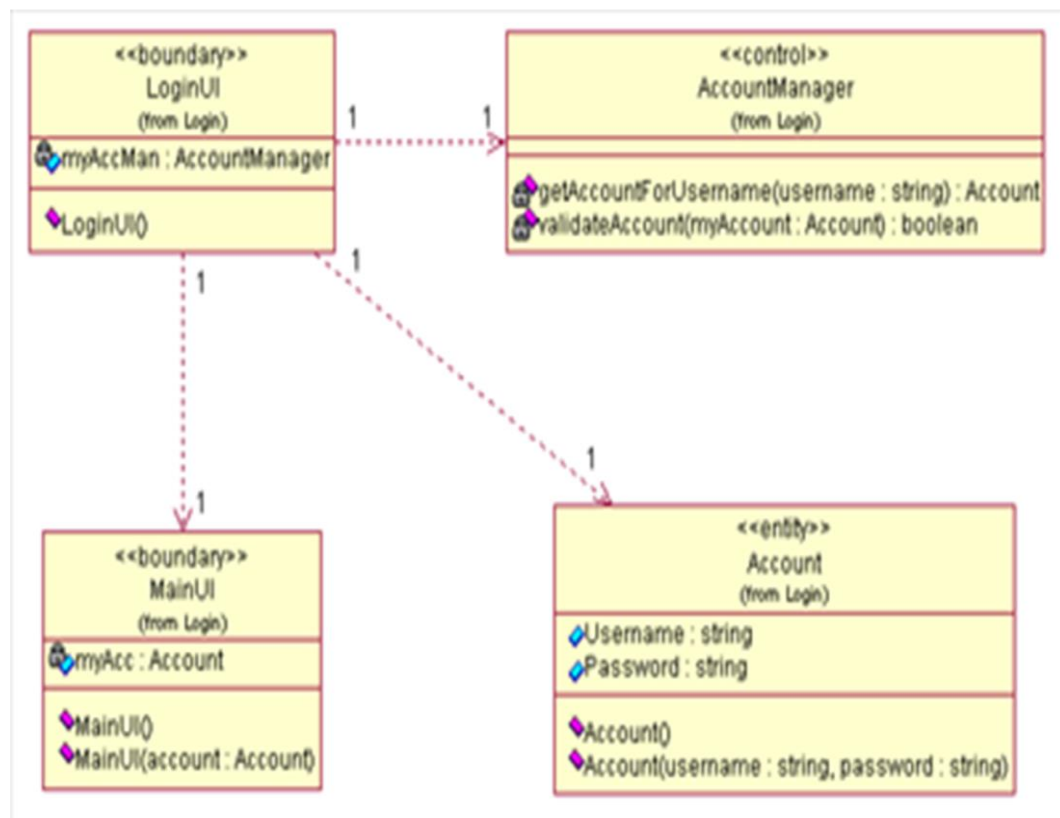
### II.12.2. Class Diagram

*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. (Jurnal Informatika Mulawarman ; Haviluddin ; 2011 : 3). Simbol-simbol yang digunakan dalam *class diagram*, yaitu :

Tabel II.6. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Havaluddin ; 2011 : 3)



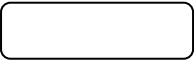
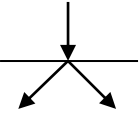
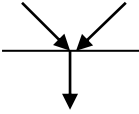
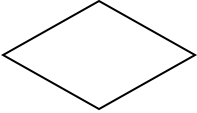

Gambar II.10. Contoh *Class Diagram*

(Sumber : Havaluddin ; 2011 : 3)

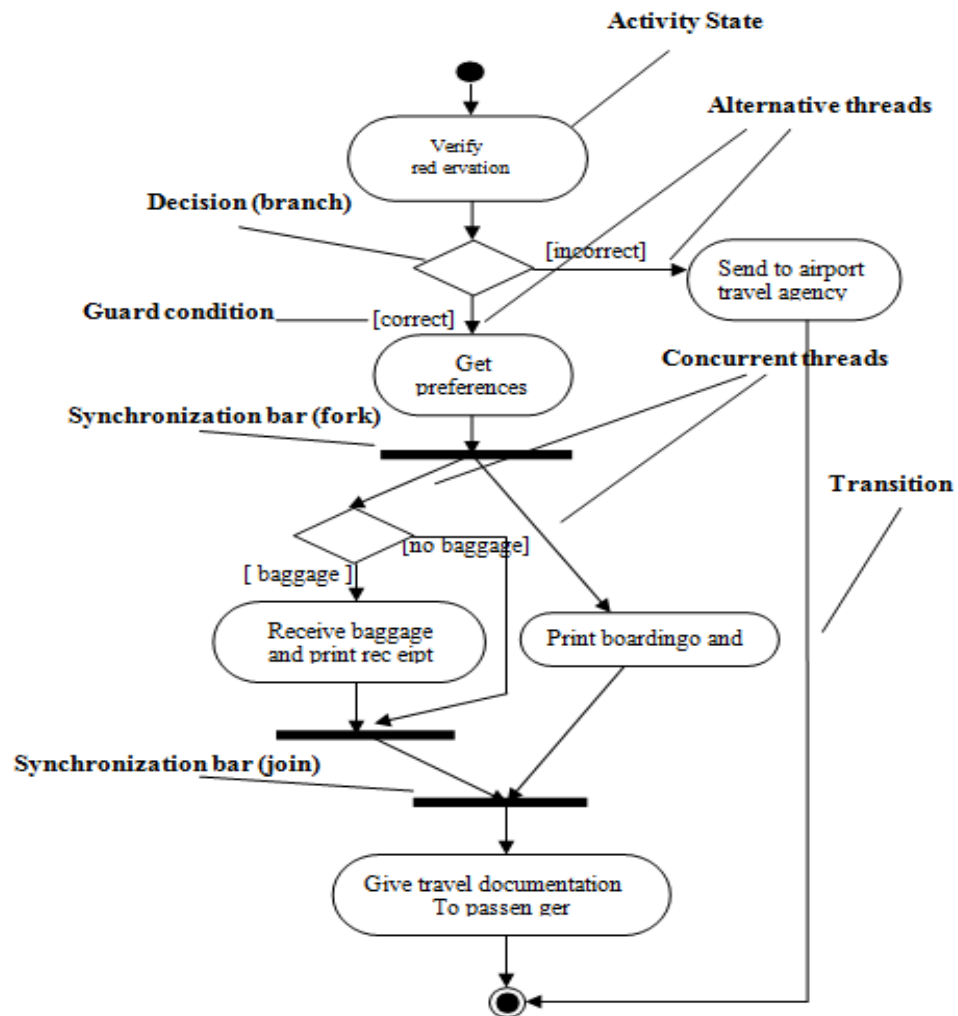
### II.12.3. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktifitas. ( Havaludin ; 2011 : 4). Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

**Tabel II.5. Simbol Activity Diagram**

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> , (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Havaluddin ; 2011 : 4)



**Gambar II.11. Contoh Activity Diagram**

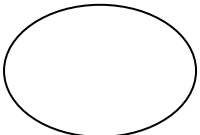
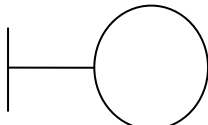
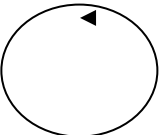

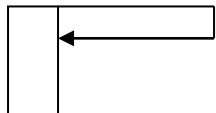


(Sumber : Havaluddin ; 2011 : 4)

#### II.12.4. Sequence Diagram

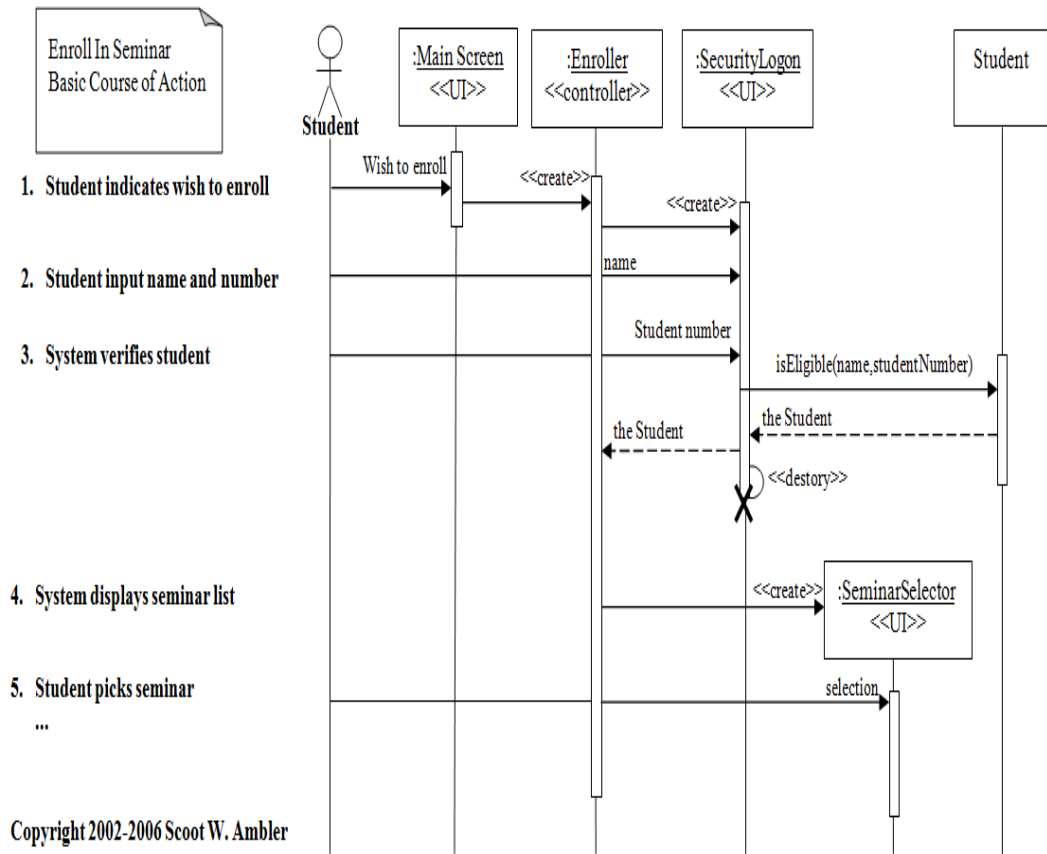
*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.

(Haviluddin ; 2011 : 5). Simbol-simbol yang digunakan dalam *sequence diagram* yaitu :

**Tabel II.7 Simbol Sequence Diagram**

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih faktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control Class</i> , objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Haviluddin ; 2011 : 5)



**Gambar II.12. Contoh Sequence Diagram**

(Sumber : Havaluddin ; 2011 : 5)