

## **BAB II**

### **LANDASAN TEORI**

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

Menurut Husain dan Wibowo (2002) Sistem Informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi.

Menurut Azhar Susanto (2008) Sistem informasi adalah kumpulan dari subsistem apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna (Jurnal Capacity STIE AMKOP Makassar ; Rizan Machmud ; 2013 : 411).

#### **II.2. 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.2.1. 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.2.2. Ciri-Ciri Sistem Pendukung Keputusan**

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

- a. Banyak pilihan/alternatif.
- b. Ada kendala atau surat.

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

### **II.2.3. Fase Dalam Sistem Pendukung Keputusan**

Tiga fase dalam pengambilan keputusan yaitu :

a. *Intelligence*

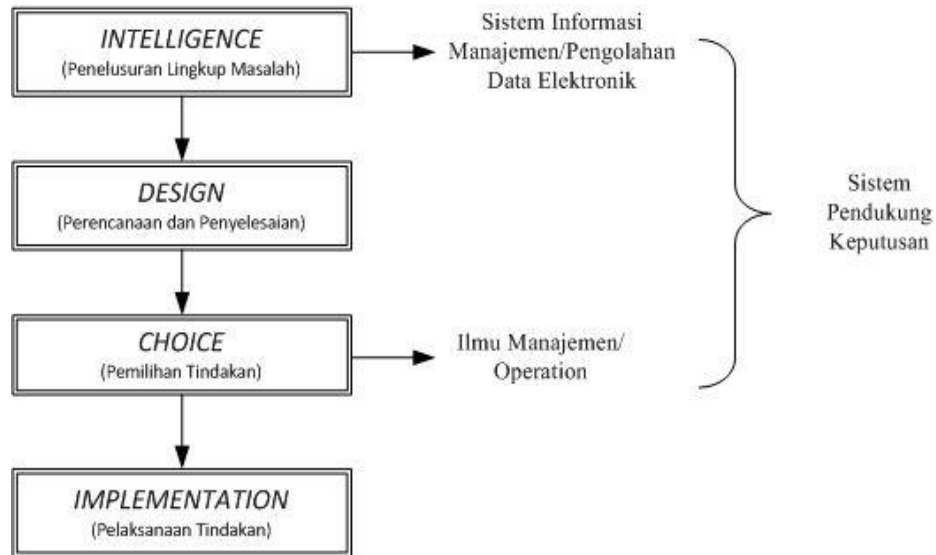
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

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 S.kom, M.kom ; 2014 : 2-3).



**Gambar II.1. Fase Proses Pengambilan Keputusan**  
 (Sumber : Dicky Nofriansyah S.kom, M.kom ; 2014 : 2-3)

#### II.2.4. 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 S.kom, M.kom ; 2014 : 3-4).

### **II.3. *Promethee***

*Promethee (Preference Ranking Organization Method for Enrichment Evaluation)* adalah satu dari beberapa metode penentuan urutan atau prioritas dalam analisis multikriteria. Metode ini dikenal sebagai metode yang efisien dan simple ,tetapi juga yang mudah diterapkan dibanding dengan metode lain untuk menuntaskan masalah multikriteria. Metode ini mampu mengakomodir kriteria

pemilihan yang bersifat kuantitatif dan kualitatif. Masalah utamanya adalah kesederhanaan, kejelasan dan kestabilan. Dugaan dari dominasi kriteria yang digunakan dalam promethee adalah penggunaan nilai dalam hubungan *outranking* (Jurnal Pelita Informatika Budi Dharma ; Nanda Abdurrahman Wahid ; 2014 :93).

Metode *Promethee* (*Preference Ranking Organizational Methods For Enrichment Evaluations*) adalah metode *outranking* yang menawarkan cara yang fleksibel dan sederhana kepada *user* untuk menganalisis masalah-masalah multikriteria (Jurnal Sarjana Teknik Informatika ; Cindra Onggo ; 2013 : 142).

### II.3.1. Fungsi Preferensi Dalam *Promethee*

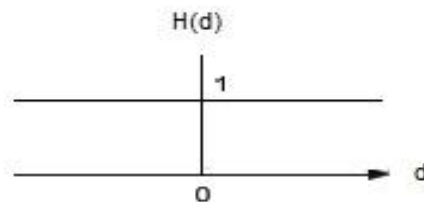
Dalam *promethee* disajikan enam bentuk fungsi preferensi kriteria. Hal ini tentu saja tidak mutlak, tetapi bentuk ini cukup baik untuk beberapa kasus. Untuk memberikan gambaran yang lebih baik terhadap area yang tidak sama, digunakan fungsi selisih nilai kriteria antara alternative  $H(d)$  dimana hal ini mempunyai hubungan langsung pada fungsi preferensi  $P$  (Jurnal Telematika ; Bambang Yuwono dkk ; 2011 ; 64). Tipe preferensi yang digunakan dalam *promethee* adalah sebagai berikut :

#### 1. Kriteria Biasa

$$H(d) \begin{cases} 0 & \text{jika } d = 0 \\ 1 & \text{jika } d \neq 0 \end{cases} \quad (1)$$

dimana  $d =$  selisih nilai kriteria  $\{d = f(a) - f(b)\}$  Pada kasus ini tidak ada beda (sama penting) antara  $a$  dan  $b$  jika hanya jika  $f(a) = f(b)$ . Apabila nilai kriteria pada masing-masing alternatif memiliki nilai berbeda,

pembuat keputusan membuat preferensi mutlak untuk alternatif yang memiliki nilai lebih baik (Jurnal Sarjana Teknik Informatika ; Cindra Onggo ; 2013 : 142).



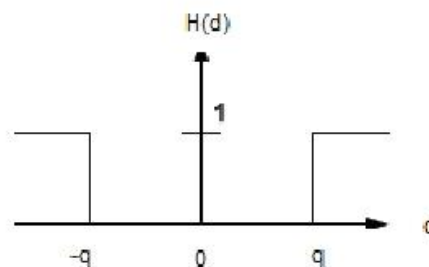
**Gambar II.2. Kriteria Biasa**

(Sumber : Cindra Onggo ; 2013 : 142)

## 2. Kriteria *Quasi*

$$H(d) \begin{cases} 0 & \text{jika } -q \leq d \leq q \\ 1 & \text{jika } d < -q \text{ atau } d > q \end{cases} \quad (2)$$

Alternatif memiliki preferensi yang sama penting jika selisih dari masing-masing alternatif untuk kriteria tertentu tidak melebihi nilai  $q$ . Apabila selisih masing-masing alternatif melebihi nilai  $q$ , maka bentuk preferensi mutlak (Jurnal Sarjana Informatika ; Cindra Onggo ; 2013 : 142).



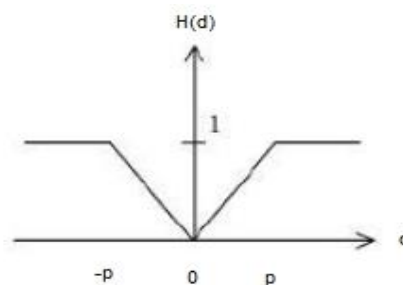
**Gambar II.3. Kriteria *Quasi***

(Sumber : Cindra Onggo ; 2013 :142)

### 3. Kriteria dengan Preferensi Linier

$$H(d) \begin{cases} d/p & \text{jika } -p \leq d \leq p \\ 1 & \text{jika } d < -p \text{ atau } d > p \end{cases} \quad (3)$$

Selama nilai selisih memiliki nilai lebih rendah dari  $p$ , maka preferensi dari pembuat keputusan meningkat secara linier dengan nilai  $d$ . Jika nilai  $d$  lebih besar dibandingkan dengan nilai  $p$ , maka terjadi preferensi mutlak (Jurnal Sarjana Informatika ; Cindra Onggo ; 2013 : 143).



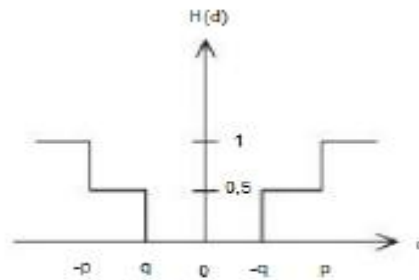
**Gambar II.4. Kriteria Preferensi Linier**

(Sumber : Cindra Onggo ; 2013 : 143)

### 4. Kriteria Level

$$H(d) \begin{cases} 0 & \text{jika } |d| \leq q \\ 0,5 & \text{jika } q < |d| \leq p \\ 1 & \text{jika } p < |d| \end{cases} \quad (4)$$

Jika berada di antara nilai  $q$  dan  $p$ , berarti  $H(d) = 0,5$  (Jurnal Sarjana Informatika ; Cindra Onggo ; 2013 : 143).



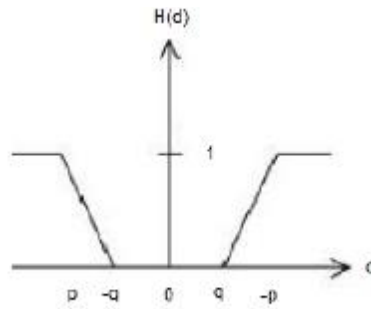
**Gambar II.5. Kriteria Level**

(Sumber : Cindra Onggo ; 2013 : 143)

5. Kriteria dengan preferensi linier dan area yang tidak berbeda

$$H(d) \begin{cases} 0 & \text{jika } |d| \leq q \\ = (|d| - q) / (p - q) & \text{jika } q < |d| \leq p \\ 1 & \text{jika } p < |d| \end{cases} \quad (5)$$

Pengambil keputusan mempertimbangkan peningkatan preferensi secara linier dan tidak berbeda, hingga preferensi mutlak dalam area antara dua kecenderungan  $q$  dan  $p$  (Jurnal Sarjana Informatika ; Cindra Onggo ; 2013 : 144).



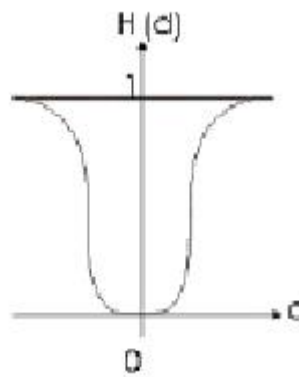
**Gambar II.6. Kriteria dengan Area Preferensi Linier Yang Tidak Berbeda**

(Sumber : Cindra Onggo ; 2013 : 144)

6. Kriteria *Gaussian*

$$H(d) = 1 - \exp \{ -d^2 / 2\sigma^2 \} \quad (6)$$

Fungsi ini bersyarat apabila telah ditentukan nilai  $\sigma$  (Jurnal Sarjana Informatika ; Cindra Onggo ; 2013 : 144).



**Gambar II.7. Kriteria *Gaussian***

(Sumber : Cindra Onggo ; 2013 : 144)

Arah dalam grafik nilai *outranking* untuk setiap node  $a$  dalam grafik nilai *outranking* ditentukan berdasarkan *leaving flow*, dengan persamaan:

$$\phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \wp(a,x) \quad (7)$$

Dimana  $\wp(a,x)$  menunjukkan *preferensi* bahwa alternatif  $a$  lebih baik dari alternatif  $x$ . *Leaving Flow* adalah jumlah dari nilai garis lengkung yang memiliki arah menjauh dari node  $a$  dan hal ini merupakan karakter pengukuran *outranking*. Selain itu, juga merupakan suatu ukuran atau nilai yang menunjukkan kekuatan dari alternatif. *Entering Flow* merupakan suatu ukuran atau nilai yang menunjukkan kelemahan dari alternatif. Secara simetris dapat ditentukan *entering flow* dengan persamaan :

$$\phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \wp(x,a) \quad (8)$$

*Net flow* menunjukkan suatu nilai total dari kekuatan dan kelemahan yang dimiliki oleh alternatif dalam penentuannya menggunakan persamaan:

$$\phi(a) = \phi^+(a) - \phi^-(a) \quad (9)$$

#### 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 :

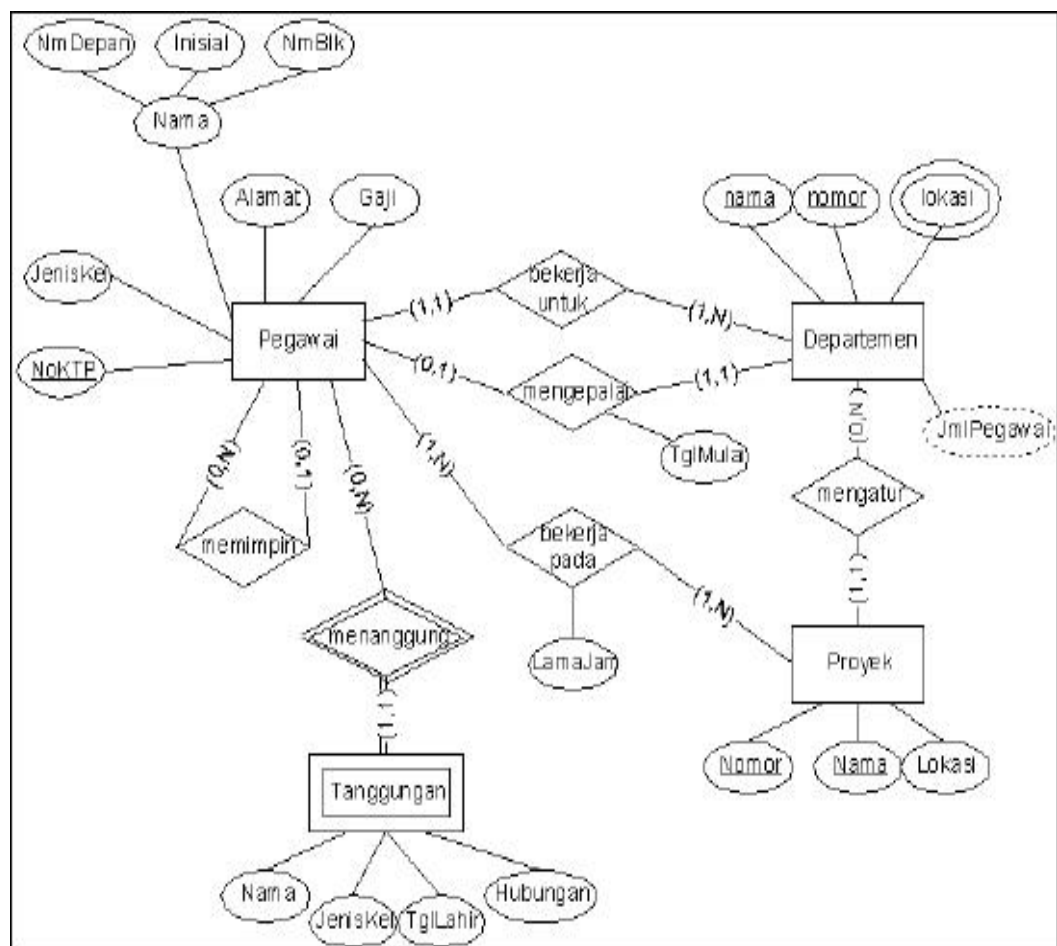
- a. Kecepatan dan kemudahan (*Speed*), melalui basis data diharapkan pengguna dapat melakukan penyimpanan, perubahan, dan menampilkan kembali dengan cepat dan mudah.
- b. Efisiensi ruang penyimpanan (*Space*), penggunaan basis data mampu mengurangi pengulangan atau redundansi data. Hal ini dapat dilakukan 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).



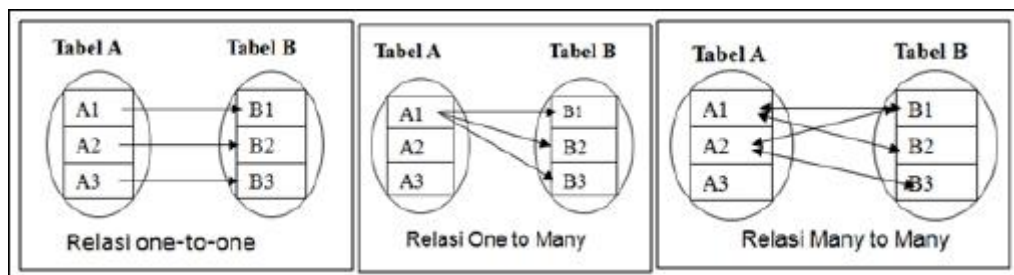
**Gambar II.8. Contoh ERD Sistem Basis Data Kepegawaian**

(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 *attribute* dari satu entitas berpasangan dengan satu *attribute* dari entitas yang direlasikan. Dalam relasi *One-to-many* (1:N) atau *many-to-one* (N:1) satu *attribute* berelasi dengan beberapa *attribute* dari entitas yang direlasikan. Dalam *Many-to-many* (M:N) satu *attribute* berelasi dengan beberapa *attribute* dari entitas yang direlasikan. Begitu pula sebaliknya (Basis Data ; Abdul Munif ; 2013 : 43).



**Gambar II.9. 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

Sumber : Sutabri (2004b:172)

### Gambar II.10. Notasi Tipe Data

(Sumber : Dea Putri Lestari ; 2013 : 36)

#### a. Notasi struktur data

Notasi struktur data digunakan untuk membuat spesifikasi elemen data. Dimana notasi yang umum digunakan adalah sebagai berikut :

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

Sumber : Sutabri (2004c:172)

### Gambar II.10. Notasi Tipe 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:

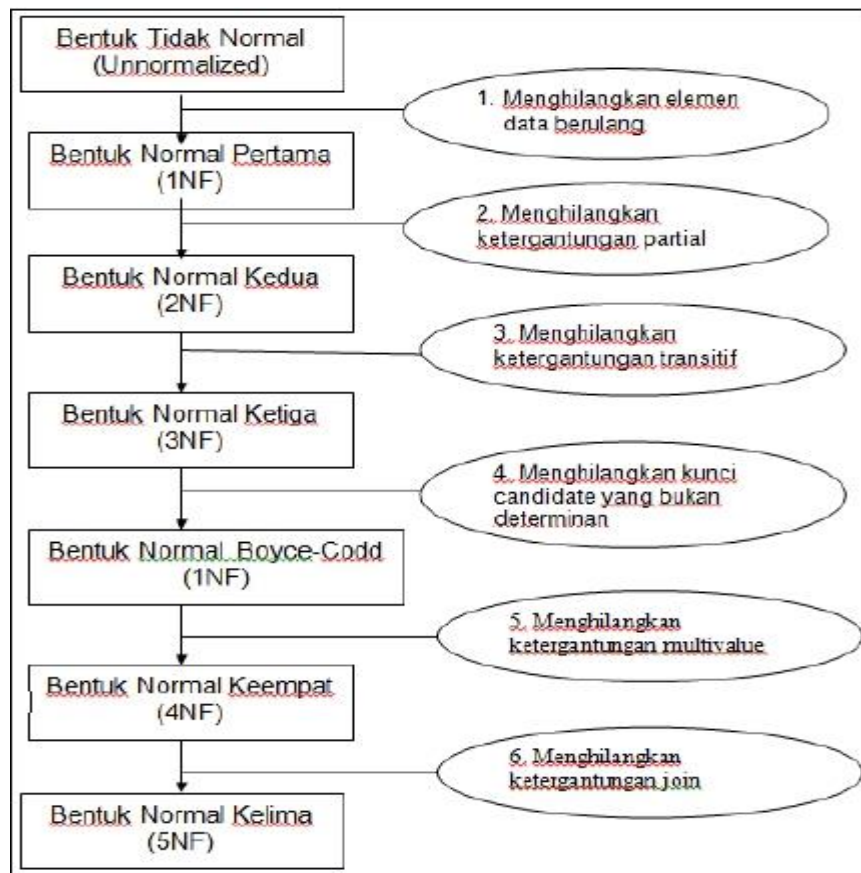
- a. Pengulangan informasi.
- b. Potensi *inkonsistensi* data pada operasi perubahan.

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.11. 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.12. 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.13. 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 attribute 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. Asam Urat**

Asam urat adalah asam yang berbentuk kristal-kristal yang merupakan hasil akhir dari metabolisme purin (bentuk turunan *nukleoprotein*), yaitu salah satu komponen asam nukleat yang terdapat pada inti sel-sel tubuh. Asam urat merupakan hasil metabolisme di dalam tubuh, yang kadarnya tidak boleh berlebih (UNESA journal of Chemistry ; Irwan Setiawan ; 2012 : 40).

Asam urat merupakan produk akhir metabolisme purin yang berasal dari metabolisme dalam tubuh, faktor endogen (genetik) dan faktor eksogen (sumber makanan). Asam urat dihasilkan oleh setiap makhluk hidup sebagai hasil dari proses metabolisme sel yang berfungsi untuk memelihara kelangsungan hidup (Kanbara,2010) (Jurnal Kesehatan Komunitas Indonesia ; Nur Lina ; 2014 : 1004).

## **II.9. Microsoft SQL Server 2008**

*SQL Server* 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. *SQL Server* adalah sebuah DBMS (*Database Management*

*System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahuluannya seperti *IBM* dan *Oracle*. *SQL Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer ; 2010 : 2).

Dalam DBMS biasanya tersedia paket bahasa yang digunakan untuk mengorganisasi basis data yang ada, yaitu :

1. *Data Defenition Language* (DDL)

*Data Definition Language* (DDL) adalah satu paket bahasa DBMS yang berguna untuk melakukan spesifikasi terhadap skema basis data. Hasil kompilasi dari DDL adalah satu set tabel yang disimpan dalam *file* khusus yang disebut *Data Directory/Dictionary*. Secara umum perintah perintah dalam DDL berhubungan dengan operasi-operasi dasar seperti membuat basis data baru, menghapus basis data, membuat tabel baru, menghapus tabel, membuat indeks, mengubah struktur tabel. Contoh perintah DDL misalnya, *Create Table*, *Create Index*, *Alter*, dan *Drop Database*.

2. *Data Manipulation Language* (DML)

*Data Manipulation Language* (DML) adalah satu paket DBMS yang memperbolehkan pemakai untuk mengakses atau memanipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat. Dengan DML dapat dilakukan kegiatan :

- a. Mengambil informasi yang tersimpan dalam basis data.
- b. Menyisipkan informasi baru dalam basis data.
- c. Menghapus informasi dari tabel.

Terdapat dua tipe DML yaitu prosedural dan non prosedural. Prosedural DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan dan bagaimana cara mendapatkannya, sedang non prosedural DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan tanpa tahu bagaimana cara mendapatkannya.

SQL merupakan kependekan dari *Structured Query Language* (Bahasa *Query* Terstruktur). SQL lebih dekat dengan DML dari pada DDL. Namun tidak berarti SQL tidak menyediakan perintah DDL. SQL lebih menekankan pada aspek pencarian dari dalam tabel. Aspek pencarian ini sedemikian penting karena di sinilah sebenarnya inti dari segala upaya kita melakukan pengelolaan data. Data dalam basis data diorganisasi sedemikian rupa dengan tujuan untuk memudahkan pencarian di kemudian hari.

Sebagai sebuah bahasa, SQL telah distandarisasi dan mengalami beberapaperubahan atau penyempurnaan. SQL muncul pertama kali pada tahun 1970 dengan nama Sequel (nama yang masih sering digunakan hingga saat ini). Standarisasi yang pertama dibuat pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan ISO (*International Standard Organization*), yang disebut SQL-86. Pada tahun 1989 SQL-86 diperbaharui menjadi SQL-89. Standar terakhir yang dibuat adalah SQL-92.

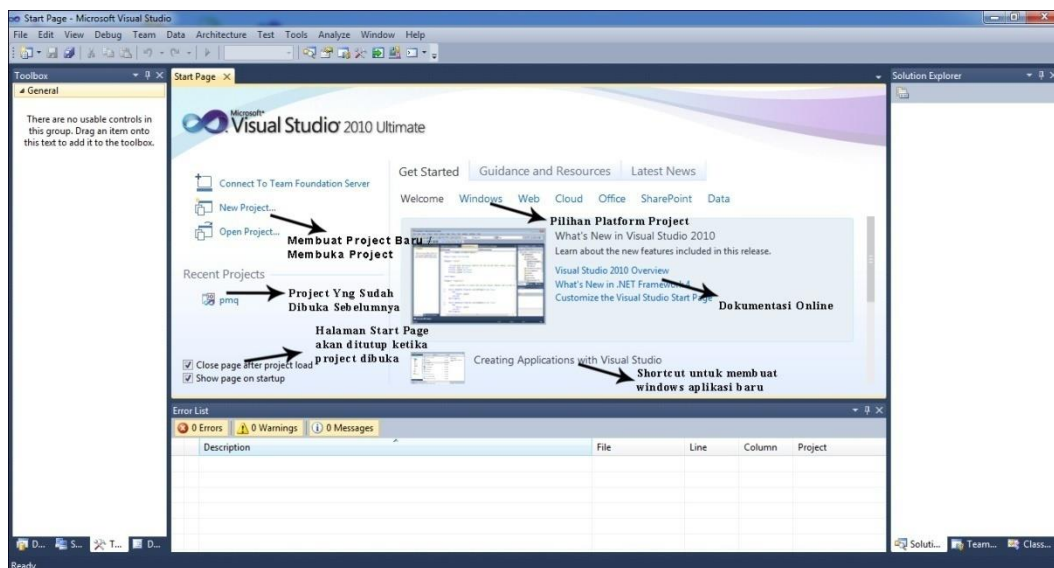
Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti *update* data pada basis data, atau menampilkan data dari basis data. Beberapa software RDBMS yang dapat menggunakan SQL, seperti : *Oracle*, *Sybase*, *Microsoft SQL Server*, *MySQL*, *Microsoft Access*, *Ingres*, dsb. Setiap *software* basis data mempunyai mungkin bahasa perintah / *sintaks* yang berbeda, namun pada prinsipnya mempunyai arti dan fungsi yang sama. Perintah utama dalam SQL adalah *select* (RPL Jilid III ; Aunur R. Mulyanto ; 2008 : 324-325).

#### **II.10. Microsoft Visual Studio 2010**

*Visual Basic* diturunkan dari bahasa *Basic*. *Visual Basic* terkenal sebagai bahasa pemrograman yang mudah untuk digunakan terutama membuat aplikasi yang berjalan diatas *platform windows*. Pada tahun 90an, *Visual Basic* menjadi bahasa pemrograman yang paling populer dan menjadi pilihan utama untuk mengembangkan program berbasis *windows*. Versi *Visual Basic* terakhir sebelum berjalan diatas *.NET Framework* adalah VB6. *Visual Basic .NET* dirilis pada bulan Februari 2002 bersamaan dengan *platform. NET Framework 1.0*. Kini sudah ada beberapa versi dari *Visual Basic* yang berjalan pada *platform .NET*, yaitu VB 2002 (VB7), VB 2005 (VB8), VB 2008 (VB9), dan yang terakhir adalah VB 2010 (VB10) yang dirilis bersamaan dengan *Visual Studio 2010*. Selain visual basic 2010, visual studio 2010 juga mendukung beberapa bahasa lain, yaitu *C#*, *C++*, *F#* (bahasa baru untuk *functional programming*), *IronPhyton* dan *IronRuby* (bahasa baru untuk *dynamic programming*) (Cepat Mahir Visual Basic ; Erick Kurniawan ; 2011 : 1).

## 1. Memulai *visual studio* 2010

Tampilan visual studio 2010 pada saat pertama kali dijalankan adalah seperti pada gambar dibawah (Cepat Mahir Visual Basic 2010 ; Erick Kurniawan ; 2011 : 10).

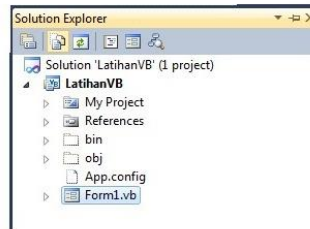


**Gambar II.14. Halaman *StartUp***

(Sumber : Erick Kurniawan ; 2011 : 10)

## 2. *Solution Explorer*

Setelah membuat project baru maka di sebelah kanan atas jendela VS 2010 akan muncul jendela *Solution Explorer*. *Solution Explorer* berisi semua *file* yang akan anda gunakan untuk membuat aplikasi. Sebuah *Solution* dapat terdiri dari beberapa *project* (Cepat Mahir Visual Basic 2010 ; Erick Kurniawan ; 2011 : 13).

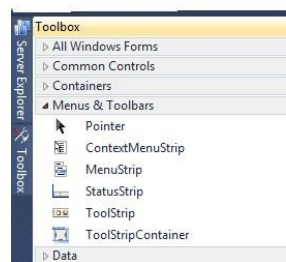


**Gambar II.15. *Solution Explorer***

**(Sumber : Erick kurniawan ; 2011 : 13)**

### 3. *Toolbox*

Secara *default*, *toolbox* mempunyai pengaturan *automatic hiding* sehingga akan tertutup jika tidak diperlukan (Cepat Mahir Visual Basic 2010 ; Erick Kurniawan ; 2011 : 14).



**Gambar II.16. *Toolbox***

**(Sumber : Erick Kurniawan ; 2011 : 14)**

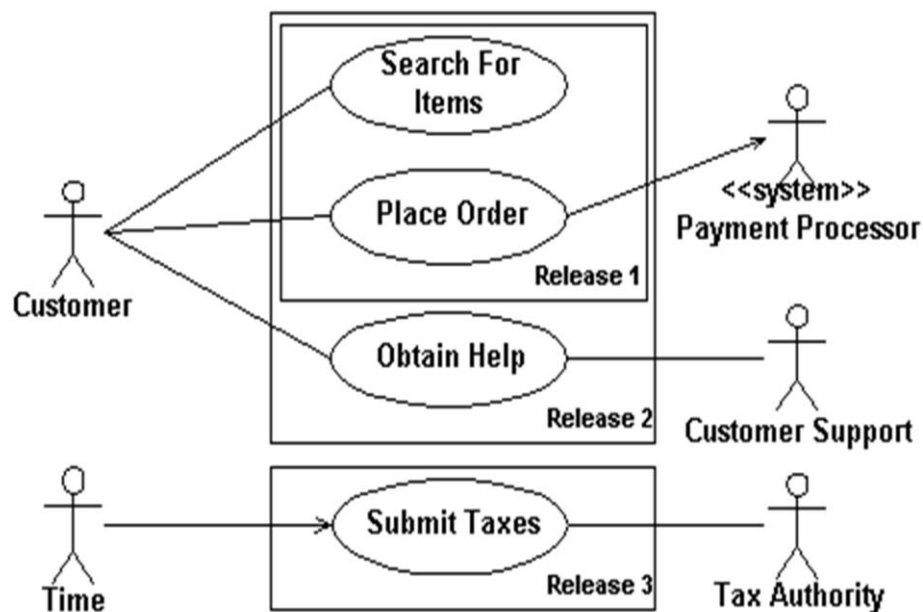
## **II.11. *Unified Modelling Language (UML)***

*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 ; Havaluddin ; 2011 : 1).

### **II.11.1. *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 ; Havaluddin ; 2011 : 4).

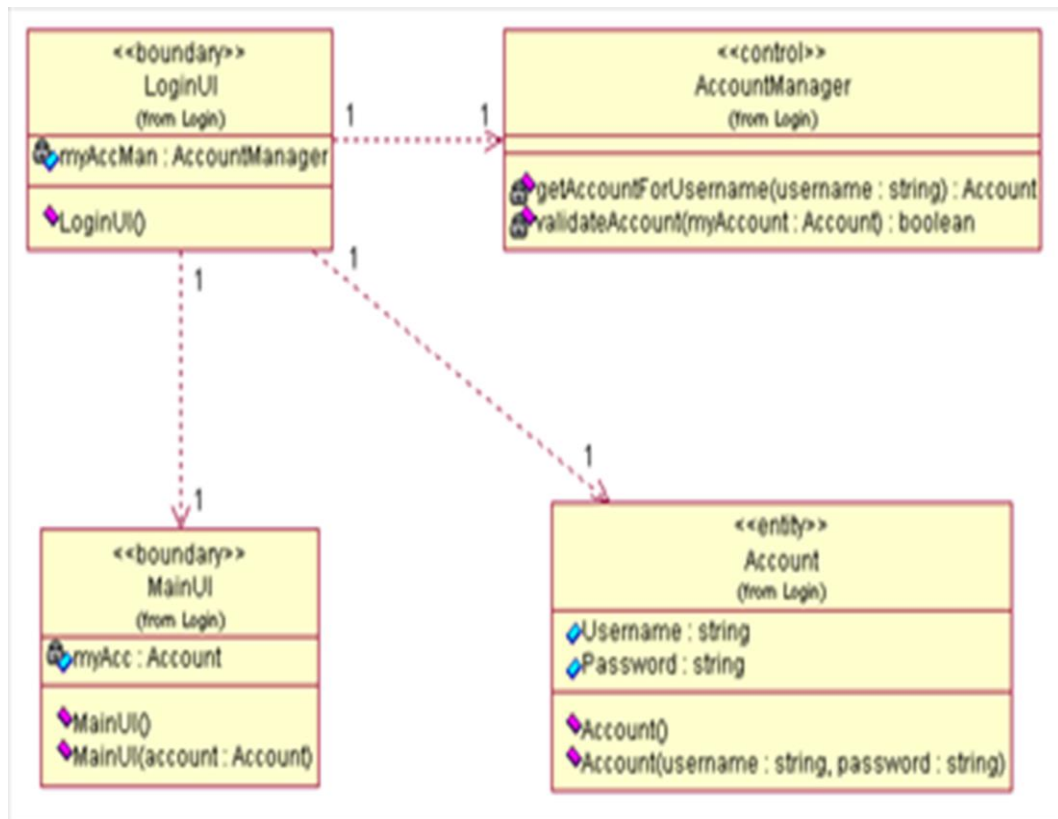


**Gambar II.17. Contoh Use Case Diagram**

(Sumber : Haviluddin ; 2011 : 4)

### II.11.2. 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).

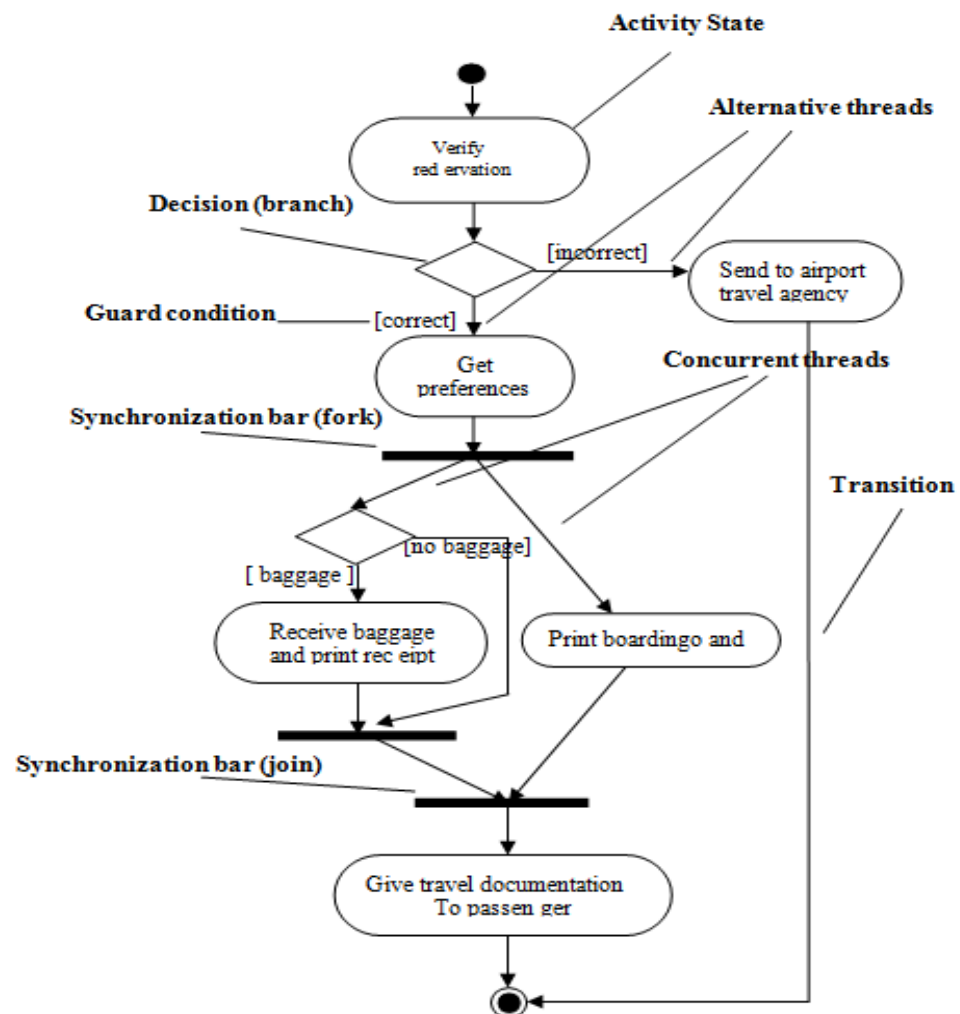


**Gambar II.18. Contoh Class Diagram**

(Sumber : Havaluddin ; 2011 : 3)

### II.11.3. 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).

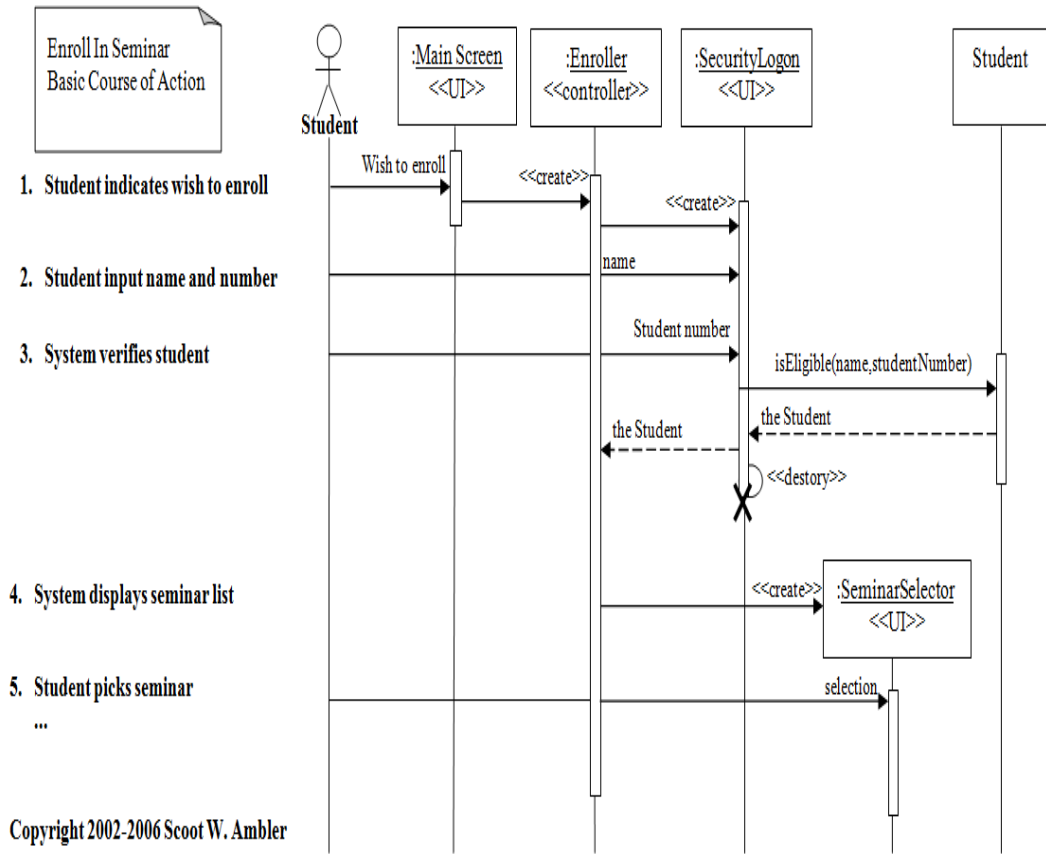


**Gambar II.19. Contoh Activity Diagram**

(Sumber : Havaluddin ; 2011 : 4)

#### II.11.4. 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*. (Havaluddin ; 2011 : 5).



**Gambar II.20. Contoh Sequence Diagram**

(Sumber : Havaluddin ; 2011 : 5)