

## **BAB II**

### **LANDASAN TEORI**

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

Sistem Pendukung Keputusan (*Decision Support System*) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter,2002) . Sistem pendukung keputusan biasanya dibangun untuk mendukung solusi atau suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan menggunakan *CBIS (Computer Based Information Systems)* yang fleksibel, interaktif, dan dapat diadaptasi yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur ( Saragih;2013: 83).

Menurut Mann dan Watson dalam buku Dadan Umar Daihan (2001), Sistem pendukung keputusan merupakan sistem berbasis komputer yang dirancang untuk mempertinggi efektifitas pengambilan keputusan dari masalah semi terstruktur. Sedangkan Maryam Alavi dan H. Albert Napier memberikan defenisi Sistem pendukung keputusan adalah sistem yang interaktif, yang membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur

dan tidak terstruktur. Dapat dikatakan bahwa Sistem pendukung keputusan bertujuan untuk menghasilkan informasi yang berkaitan langsung dengan proses pengambilan keputusan baik yang bersifat semi terstruktur maupun yang tidak terstruktur. (Pratiwi ; 2015 : 39).

Aplikasi sistem pendukung keputusan menggunakan data, memberikan antar muka pengguna yang mudah dan dapat menggabungkan pemikiran pengambilan keputusan. Sistem pendukung keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. Sistem Pendukung Keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari Sistem Pendukung Keputusan (*Decision Support Systems*) adalah sebagai berikut :

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.

5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

(Saragih;2013: 83).

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

Sistem Pendukung Keputusan (*Decision Support Systems*) pertama kali digunakan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision Systems*. Selanjutnya perusahaan, lembaga penelitian dan perguruan tinggi mulai melakukan penelitian dan membangun Sistem Pendukung Keputusan. Perbedaan utama antara sistem pendukung keputusan dengan sistem informasi adalah bahwa sistem informasi manajemen menghasilkan informasi yang lebih bersifat rutin dan terprogram, sedangkan sistem pendukung keputusan sudah dikaitkan dengan proses pengambilan keputusan yang spesifik. Beberapa karakteristik Sistem Pendukung Keputusan adalah sebagai berikut :

1. Sistem Pendukung Keputusan dirancang untuk membantu pengambilan keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan pengguna model-model teknik-teknik analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari atau interogasi informasi.

3. Sistem pendukung keputusan dirancang sedemikian rupa sehingga dapat digunakan atau dioperasikan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
4. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai. (Pratiwi ; 2015 : 39).

## **II.2. Metode *Analytical Hierarchy Process* (AHP)**

*Analytical Hierarchy Process*(AHP) merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi kriteria yang kompleks menjadi suatu hierarki. Menurut Saaty, Hierarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan yang diikuti level faktor, kriteria, sub kriteria dan seterusnya ke bawah hingga level terakhir dari alternative. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompok yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis. (Saragih ; 2013 : 83).

AHP dikembangkan di Wharton School of Business yang kemudian menjadi alat yang sering digunakan dalam pengambilan keputusan. AHP

merupakan proses dalam pengambilan keputusan dengan menggunakan perbandingan berpasangan (*Pair wise Comparisons*) untuk menjelaskan faktor evaluasi dan faktor bobot dalam kondisi multi faktor. Dengan demikian AHP digunakan manakala keputusan yang diambil melibatkan banyak faktor, dimana pengambilan keputusan mengalami kesulitan dalam membuat bobot setiap faktor tersebut.

AHP memecahkan suatu situasi yang kompleks, tidak terstruktur ke dalam beberapa komponen dalam susunan yang hierarki dengan memberi nilai subjektif tentang pentingnya setiap variable secara relatif dan menetapkan variable secara relatif dan menetapkan variable mana yang memiliki prioritas paling tinggi guna mempengaruhi hasil pada situasi tersebut. AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihak yang terlibat dalam pengambilan keputusan. Menurut Yahya dalam buku (Suryadi dan Ramdhani, 2002) adakalanya timbul masalah keputusan yang dirasaka dan diamati perlu diambil secepatnya tetapi variasinya rumit sehingga tidak mungkin datanya dicatat secara numeric, hanya secara kualitatif saja yang diukur yaitu berdasarkan persepsi pengalaman dan intuisi. Namun tidak menutup kemungkinan bahwa model-model lainnya ikut dipertimbangkan pada saat proses pengambilan keputusan dengan pendekatan AHP khususnya dalam memahami para pengambil keputusan individual pada saat proses penerapan pendekatan ini. (Berutu ; 2015 : 96,97,98)

### **II.2.1. Kelebihan Metode AHP**

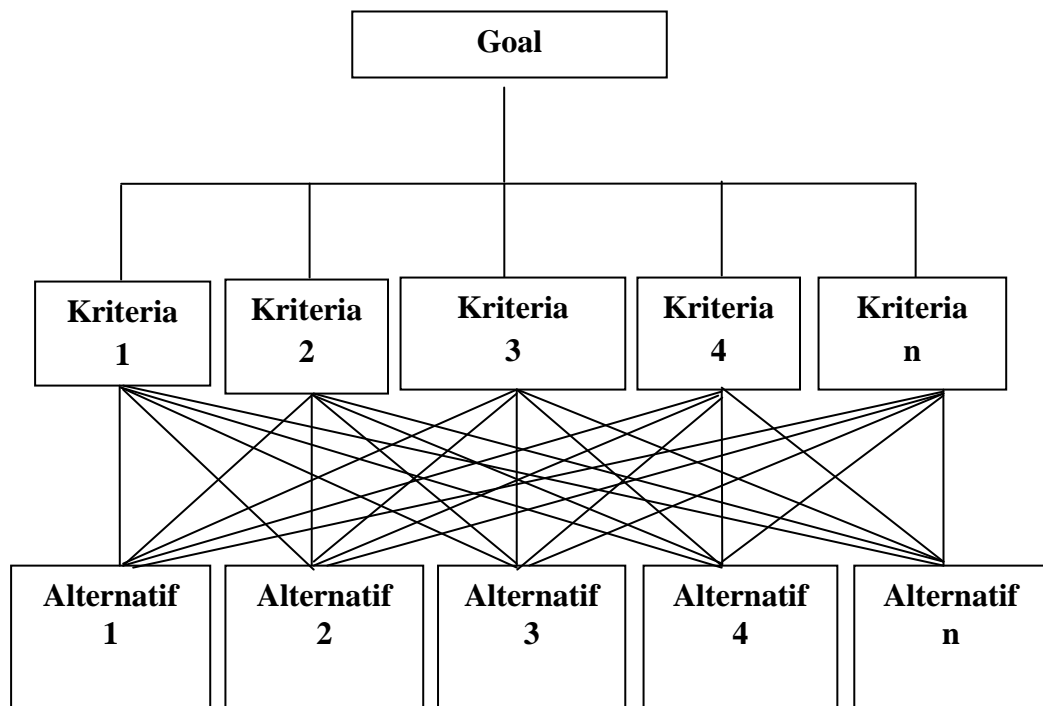
Kelebihan dari model AHP dibandingkan dengan model pengambilan keputusan yang lain terletak pada kemampuannya untuk memecahkan masalah yang multi objektivitas dengan multi kriteria. Kebanyakan model yang sudah ada memakai *single objectives* dengan multikriteria. Kelebihan model AHP ini lebih disebabkan oleh fleksibilitasnya yang tinggi terutama dalam pembuatan hirarki. Sifat fleksibel tersebut membuat model AHP dapat menangkap beberapa tujuan dan beberapa kriteria sekaligus dalam sebuah model atau sebuah hirarki (Hidayat ; 2014 : 10).

### **II.2.2. Prinsip Dasar AHP**

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami. Diantaranya adalah sebagai berikut :

1. Membuat Hierarki

Sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hieraki dan menggabungkannya atau mensintesisnya.



**Gambar II.1. Struktur Hierarki Analytical Hierarchy Process (AHP)**

(Sumber : Syahrani, dkk ; 2013 : 29)

## 2. Penilaian Kriteria dan Alternatif

Kriteria dan Alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (1998) untuk berbagai persoalan skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan defenisi pendapat kualitatif dan skala perbandingan Saaty bisa diukur menggunakan sebagai berikut :

**Tabel II.1. Skala Penilaian Perbandingan Pasangan**

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika elemen $i$ memiliki salah satu angka dari skala perbandingan 1 sampai 9 yang telah ditetapkan oleh Saaty ketika dibandingkan dengan elemen $j$ , maka $j$ memiliki kebalikannya ketika dibandingkan dengan elemen $i$

(Sumber : Kusrini ; 2007 : 134)

### 3. *Synthesis Of Priority* (Menentukan Prioritas)

Untuk setiap kriteria dan alternative perlu dilakukan perbandingan berpasangan (*Pair wise Comparisons*). Nilai-nilai perbandingan relative dari seluruh alternatif kriteria bisa disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian permasalahan matematika.

### 4. *Logical Consistency* (Konsistensi Logis)

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua,

menyangkut tingkat hubungan antarobjek yang didasarkan pada kriteria tertentu. (Berutu ; 2015 : 98-99)

### **II.2.3. Prosedur AHP**

Pada dasarnya, prosedur atau langkah-langkah dalam metode AHP meliputi ( Berutu ; 2015 : 98-99) :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan sasaran sistem secara keseluruhan pada level atas.
2. Menentukan prioritas elemen.
  - a. Langkah pertama adalah membuat perbandingan pasangan yaitu membandingkan elemen secara berpasangan sesuai dengan kriteria yang diberikan.
  - b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.
3. Sintesis hal-hal yang dilakukan dalam langkah ini adalah :
  - a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
  - b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
  - c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

#### 4. Mengukur konsistensi

Dalam pembuatan keputusan penting untuk mengetahui seberapa baik konsistensi yang ada karena tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :

- a. Mengalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua dan seterusnya.
- b. Menjumlahkan setiap baris.
- c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
- d. Menjumlahkan hasil bagi diatas dengan banyaknya elemen yang ada hasilnya disebut  $\lambda$  maks.

#### 5. Menghitung *Consistency Index* (CI) dengan rumus

$$CI = (\lambda_{maks} - n) / n \dots\dots\dots ( \text{Rumus I} )$$

Dimana n = banyaknya elemen

#### 6. Menghitung Rasio Konsistensi/*Consistency Ratio* (CR) dengan rumus :

$$CR = CI / RC \dots\dots\dots ( \text{Rumus II} )$$

Dimana : CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Index Random Consistency*

Keterangan :

$\lambda$  maks = Maximum Eigen Value

N = Ukuran Matriks

CR = Consistency Ratio

CI = Consistency Index

IR = Index Random Consistency

7. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian dari data judgment harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1 maka hasil perhitungan bisa dinyatakan benar. Penentuan indeks random konsistensi mengacu pada tabel dibawah ini

**Tabel II.2. Daftar Indeks Random Konsistensi**

<b>Ukuran Matriks</b>	<b>Nilai IR</b>
1,2	0,00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

(Sumber : Kusriani ; 2007 : 136)

### II.3. Pengertian Basis Data (*Database*)

Sebuah basis data adalah kumpulan data yang saling berelasi dan merupakan himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah. Basis data juga dapat didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu untuk memenuhi kebutuhan. Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. (Kusrini : 2007 ; 2).

Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengkonsolidasikan berbagai catatan yang terlebih dahulu disimpan file-file terpisah kedalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi.

### II.4. ERD (*Entity Relationship Diagram*)

Perancangan basis data dengan menggunakan model *entity Relationship* adalah dengan menggunakan ERD. Terdapat 3 notasi dasar yang bekerja pada model E-R yaitu : *entity sets*, *relationship sets*, dan *attributes*. Sebuah entity adalah sebuah “benda” atau objek di dunia nyata yang dapat dibedakan dari semua objek lainnya. (Kusrini : 2007 ; 21)

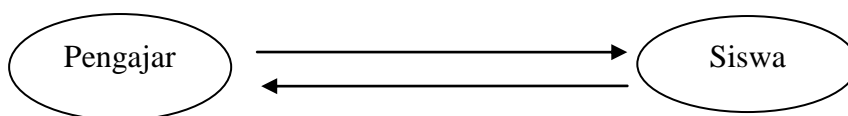
Pada dasarnya ERD adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD. ERD digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya.

Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut.

Menurut Tata Sutabri (2005;208) Relasi antara 2 file atau 2 tabel dapat dikategorikan menjadi 3 macam. Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada di dalam 1 atau 2 file, yaitu adalah seperti berikut :

1. *One to One Relationship 2 File*

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.



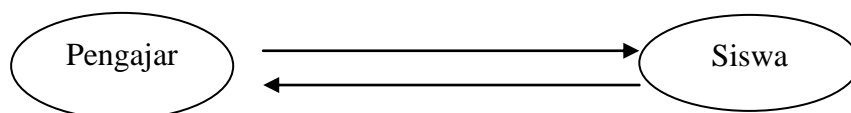
**Gambar II.2. One To One Relationship 2 File**

**(Sumber : Sutabri 2005;209)**

2. *One to Many Relationship 2 File*

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula berbalik, banyak lawan satu.

Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antarkeduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

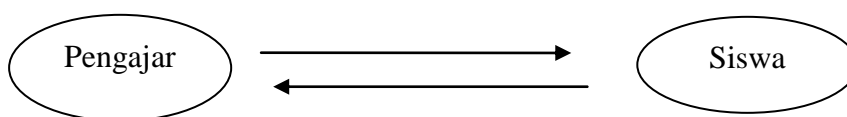


**Gambar II.3. *One To Many Relationship 2 File***

**(Sumber : Sutabri 2005;209)**

3. *Many to many Relationship 2 File*

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antarkeduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.



**Gambar II.4. *Many To Many Relationship 2 File***

**(Sumber : Sutabri 2005;209)**

## II.5. Kamus Data

Seperti halnya dengan kamus bahasa yang berfungsi menjelaskan lebih detail suatu kata maupun kalimat, kamus data yang digunakan dalam analisis struktur dan desain sistem informasi juga merupakan suatu katalog yang menjelaskan lebih detail tentang DFD yang mencakup proses, data flow dan data store. Kamus data dapat digunakan pada metodologi objek, kamus data dapat menjelaskan lebih detail atribut maupun metode atau service suatu objek. Secara defenisi, kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan arus data yang ada pada data flow diagram. Arus data yang ada di DFD bersifat global dan hanya menunjukkan nama arus datanya saja. Kamus data atau data dictionary harus dapat mencerminkan keterangan yang jelas tentang data yang dicatat. (Sutabri:2005;170)

Sumber kamus data, yaitu :

1. Data Store (File-File)
2. Data Flow (Aliran Data)
3. Data Elemen yang dinyatakan dalam spesifikasi data dan berasal dari file.

**Tabel II.3. Notasi Kamus Data**

Notasi	Keterangan
=	Terdiri dari
+	Dan
()	Pilihan (ya atau tidak)

{ }	Pengulangan proses
[ ]	Pilih salah satu pilihan
**	Catatan
@	Petunjuk
	Pemisah pilihan dalam tanda []

(Sumber : Sutabri ; 172)

Contoh penggunaan notasi untuk data penjualan.

@NomorFaktur

NamaPelangan

Alamat = Jalan + Kota + Kode Pos

(Telepon)

SistemPembayaran (Kas | Kredit)

Keterangan \* Mencatat Informasi Tambahan

NamaStock

UnitJual

HargaJual

## II.6. Normalisasi

Normalisasi adalah bagian perancangan basis data. Tanpa Normalisasi, sistem basis data menjadi tidak akurat. Dengan normalisasi dapat mendesain database relasional yang terdiri dari tabel :

1. Berisi data yang diperlukan
2. Memiliki sesedikit mungkin redundansi
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan
4. Mengefisienkan *update*
5. Menghindari kemungkinan kehilangan data secara tidak disengaja.

Alasan utama dari normalisasi database minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya *insertion anomalies*, *delete anomalies*, dan *update anomalies*. Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. (Kusrini:2007;40)

Teknik normalisasi juga merupakan suatu teknik yang menstrukturkan data dalam cara tertentu untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam database. Proses normalisasi menghasilkan struktur record yang konsisten secara logic yang mudah untuk dimengerti dan sederhana dalam pemeliharaannya.

Bentuk-bentuk Normalisasi menurut Tata Sutabri adalah sebagai berikut :

1. Bentuk tidak normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi

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

Bentuk normal pertama mempunyai ciri yaitu bahwa setiap data dibentuk dalam *flat file*. Data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa *atomic value*. Tiap *field* hanya mempunyai satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukanlah pecahan kata sehingga memiliki arti yang lain.

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

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama (*primary key*). Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan kunci *field*. Kunci *field* haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

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

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

5. *Boyce-Codd Normal Form* (BCNF)

*Boyce-Codd Normal Form* mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi *Boyce-Codd Normal Form*, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey*.

## II.7. SQL Server 2008

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data rasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

SQL terdiri dari dua bahasa yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD) namun secara umum implementasi setiap bahasa ini memiliki bentuk standar yang ditetapkan oleh ANSI.

### 1. *Data Definition Language* (DDL)

DDL digunakan untuk mendefinisikan mengubah serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

Secara umum DDL yang digunakan adalah :

- a. *CREATE* untuk membuat objek baru.
- b. *USE* untuk menggunakan objek.
- c. *ALTER* untuk mengubah objek yang sudah ada.
- d. *DROP* untuk menghapus objek.

## 2. *Data Manipulation Language (DML)*

*DML* digunakan untuk memanipulasi data yang ada dalam suatu tabel.

Perintah-perintah yang umum dilakukan adalah :

- a. *SELECT* untuk menampilkan data.
- b. *INSERT* untuk menambahkan data baru.
- c. *UPDATE* untuk mengubah data yang sudah ada.
- d. *DELETE* untuk menghapus data. (Adelia, Setiawan ; 2011 : 115)

### **II.8. Microsoft Visual Basic 2010**

Pada akhir tahun 1999, Teknologi .NET diumumkan oleh Microsoft memosisikan teknologi tersebut sebagai *platform* untuk membangun XML Web Services. XML Web Services memungkinkan aplikasi tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet. Visual Basic.Net adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada sistem komputer apa pun dan dapat mengambil data dari server dengan tipe apa pun asalkan terinstal .NET Framework.

Visual Studio pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows. Visual Studio adalah inkarnasi dari bahasa Visual Basic

yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat tinggi lainnya seperti C++ (Christopher Lee,2014;1).

## **II.9. *Unified Modeling Language (UML)***

*Unified Modeling Language (UML)* adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin:2011;1).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda sehingga dapat dikatakan belum memiliki standarisasi. Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group (OMG)* maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rational Software Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rational. (Haviluddin:2011;1)

UML (*Unified Modeling Language*) adalah bahasa untuk spesifikasi, visualisasi, pembangunan dan dokumentasi sistem perangkat lunak. Pada

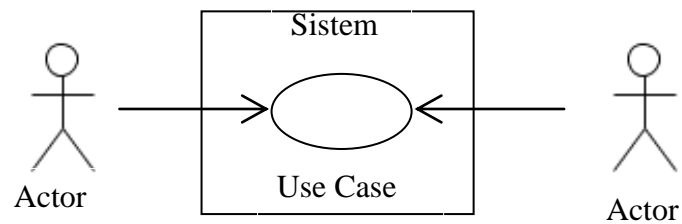
perancangan UML, sistem didefinisikan sebagai sekumpulan objek yang memiliki atribut dan metode. UML sebagai metode pemodelan berorientasi objek.

Pada level atas, cara pandang dalam UML dapat dibagi menjadi tiga area: *structural clasification*, *dynamic behavior*, dan *model management*. Klasifikasi struktural menjelaskan suatu objek dalam sistem dan hubungan antar objek tersebut. Pembahasan di dalamnya menyangkut *class*, *use case*, komponen dan *node*. *Dynamic behavior* menjelaskan perilaku sistem dari waktu ke waktu. Termasuk dalam *dynamicbehavior view* adalah *state machine view*, *activityview*, dan *interaction view*. *Model management* menjelaskan pengorganisasian model secara hierarkis. (Abdurohman, dkk:2010;93).

Menurut (Hamim Tohari:2014) Diagram-diagram yang termasuk di dalam UML, yaitu :

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

*Use Case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor. *Use Case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. Diagram *Use Case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use Case* menyatakan sebuah aktivitas atas pekerjaan tertentu. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

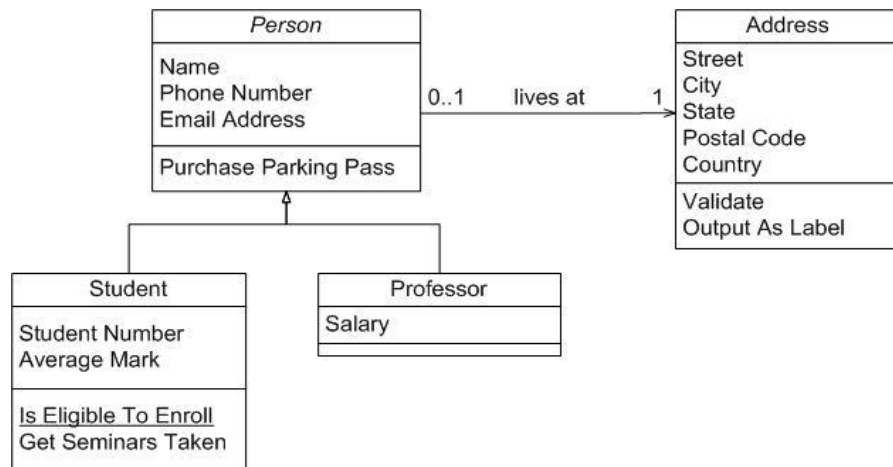


**Gambar II.5. Use Case Diagram**

**(Sumber : Havilludin : 2011;4)**

## 2. Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. *Class Diagram* menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi dan hubungan antar 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.




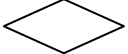



**Gambar II.6. Class Diagram**

(Sumber : Havilludin : 2011;3)

### 3. Activity Diagram

digunakan untuk menggambarkan aktifitas-aktifitas, objek, *state*, transisi, *state* dan *event*. *Activity diagram* digunakan untuk menganalisis perilaku sistem dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi di antara mereka satu sama lain. Dengan kata lain diagram alur kerja atau *Activity Diagram* menggambarkan perilaku sistem untuk aktivitas.

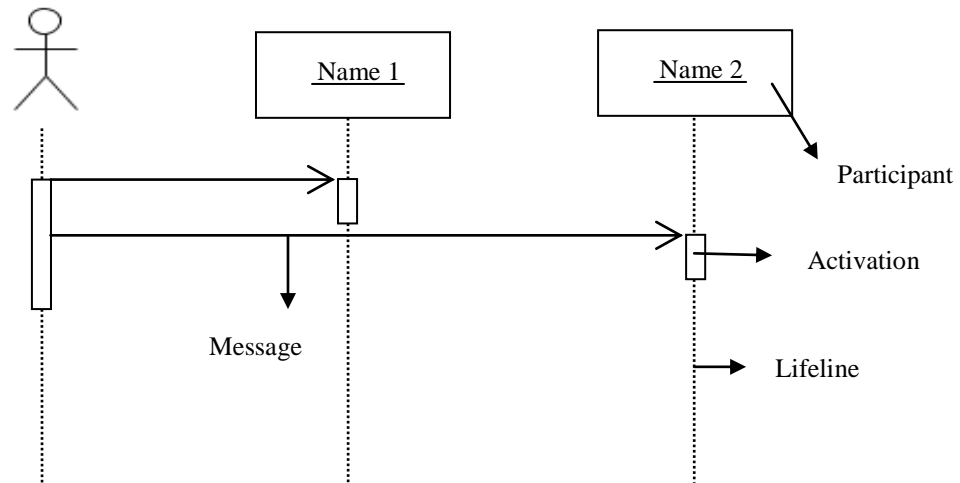
Gambar	Nama
	Titik awal
	Titik akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu

**Gambar II.7. Activity Diagram**

(Sumber : Havilludin (2011;5))

#### 4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Dalam UML, objek, pada diagram sequence digambarkan dengan segi empat yang berisi nama dari objek yang digarisbawahi. Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.



**Gambar II.8. Sequence Diagram**

**(Sumber : Havilludin (2011;5))**