

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Definisi Data**

Data adalah segala fakta, angka, atau teks yang dapat diproses oleh komputer. Saat ini, akumulasi pertumbuhan jumlah data berjalan dengan cepat dalam format dan basis data yang berbeda. (Eko Prasetyo ; 2012 : 4).

Data – data tersebut antara lain :

- a. Data operasional atau transaksional, sepperti penjualan, inventaris, penggajian, akuntansi, dan sebagainya.
- b. Data non operasional, seperti industry penjualan (supermarket), peramalan, dan data ekonomi makro.
- c. Metadata adalah data mengenai data itu sendiri, seperti desain logika basis data definisi kamus data.

#### **II.2. Data Mining**

##### **II.2.1. Pengertian Data Mining**

Data mining merupakan proses yang menggunakan teknik statistic, matematika, kecerdasan buatan dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar. (Turban et al, 2005 ). (Jurnal DASI;Emha Taufik Lutfi: 2009).

Menurut Gartner Group, *data mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005). (Jurnal DASI;Emha Taufik Lutfi:2009).

Data mining juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Pramudiono, 2007).(Jurnal SAINTIKOM;Dian Wirdasari,dkk:2011:139).

Definisi lain diantaranya adalah pembelajaran berbasis induksi (*induction based learning*) adalah proses pembentukan definisi – definisi konsep umum yang dilakukan dengan cara mengobservasi contoh – contoh spesifik dari konsep – konsep yang akan dipelajari. *Knowledge Discovery in Databases* (KDD) adalah penerapan metode saintifik pada data mining. Dalam konteks ini data mining merupakan satu langkah dari proses KDD. (Fajar Astuti Hermawati; 2013: 3).

Data mining merupakan proses iterative dan interaktif untuk menemukan pola atau model baru yang sah (sempurna), bermanfaat dan dapat dimengerti dalam suatu database yang sangat besar (*massive databases*). (Fajar Astuti Hermawati; 2013: 3).

## **II.2.2. Alasan Melakukan Data Mining**

Alasan mengapa melakukan data mining dari sudut pandang komersial karena :

1. Meledaknya volume data yang dihimpun dan disimpan dalam data warehouse seperti data web, e-commerce, penjualan di department-store, transaksi bank / credit card.
2. Proses komputasi yang dapat diupayakan.
3. Kuatnya tekanan kompetitif untuk dapat menyediakan yang lebih baik, layanan – layanan custom-isasi dan informasi sedang menjadi produk yang berarti. (Fajar Astuti Hermawati; 2013: 4).

### **II.2.3. Permasalahan Dalam Data Mining**

Sistem *data mining* berdasar pada basis data yang menyediakan data mentah dan ini memunculkan permasalahan dalam basis data yang cenderung dinamis, tidak lengkap, ber-noise dan besar. Permasalahan lain muncul sebagai akibat dari kecukupan dan relevansi dari informasi yang disimpan.

Basis data sering kali didesain untuk tujuan yang berbeda dari *data mining* dan kadangkala property atau atribut yang akan menyederhanakan pekerjaan pembelajaran tidak tersedia atau tidak dapat dimintai dari dunia nyata. Data yang tidak meyakinkan menyebabkan permasalahan karena jika ada atribut-atribut esensial bagi pengetahuan tentang domain aplikasi tidak ada dalam data tidak memungkinkan untuk menemukan pengetahuan yang tepat mengenai domain yang diberikan. Sebagai contoh, kita tidak dapat mendiagnosa malaria dari basis data pasien jika basis data tersebut tidak mengandung jumlah sel darah merah pasien.

Basis data biasanya dicemari oleh *error* sedemikian hingga tidak dapat diasumsikan bahwa data secara keseluruhan benar. Atribut-atribut yang ada pada subyek atau pertimbangan ukuran dapat memunculkan kesalahan atau *error* sedemikian hingga beberapa contoh mungkin menjadi mis-klasifikasi. Error dalam salah satu nilai atribut atau informasi kelas dikenal sebagai noise. Secara nyata ada kemungkinan kita perlu sekali untuk menghilangkan noise dari informasi klasifikasi saat hal ini mempengaruhi akurasi aturan yang dibangkitkan secara keseluruhan.

Data yang hilang dapat dibenahi dengan sistem penemuan dalam berbagai cara, seperti :

1. Secara sederhana dengan mengabaikan nilai-nilai yang hilang.
2. Menghilangkan *record* yang berhubungan.
3. Menebak nilai yang hilang dari nilai-nilai yang diketahui.
4. Memperlakukan data yang hilang sebagai sebuah nilai khusus yang dimasukkan sebagai tambahan dalam domain atribut.
5. Menghitung rata-rata nilai yang hilang menggunakan teknik *Bayesian*.

Data yang mengandung *noise* dalam pengertian menjadi tidak teliti merupakan karakteristik dari semua koleksi data dan secara khusus cocok untuk sebuah distribusi statistic seperti *Gaussian* saat nilai-nilai yang salah merupakan kesalahan masukkan data. Ketidakjelasan (*uncertainty*) menunjuk kepada kepelikan *error* dan tingkat noise dalam data. Presesi data merupakan salah satu pertimbangan penting dalam sistem penemuan.

Basis data cenderung menjadi besar dan dinamis dalam hal isinya yang selalu berubah saat informasi ditambahkan, dimodifikasi atau dihapus. Permasalahan dalam hal ini dari sudut pandang data mining adalah bagaimana menjamin bahwa aturan-aturan tersebut up-to-date dan konsisten dengan informasi paling terkini. Juga sistem pembelajaran mempunyai time-sensitive saat beberapa nilai data berubah terhadap waktu dan sistem penemuan dipengaruhi oleh ketepatan waktu dari data tersebut. (Fajar Astuti Hermawati; 2013: 13-14).

#### **II.2.4. Teknik Data Mining**

Beberapa teknik dan sifat data mining adalah sebagai berikut :

- a. *Classification (Predictive)*
- b. *Clustering (Descriptive)*
- c. *Association Rule Discovery (Descriptive)*
- d. *Sequential Pattern Discovery (Descriptive)*
- e. *Regression (Predictive)*
- f. *Deviation Detection (Predictive)* (Fajar Astuti Hermawati; 2013:14).

#### **II.2.5. Tantangan Dalam Data Mining**

Tantangan dalam data mining meliputi :

1. *Scalability*, yaitu besarnya ukuran basis data yang digunakan.
2. *Dimensionality*, yaitu banyaknya jumlah atribut dalam data yang akan diproses.

3. *Complex and Heterogeneous Data*, yaitu data yang kompleks dan mempunyai variasi yang beragam.
4. *Data Quality*, kualitas data yang akan diproses seperti data yang bersih dari noise, missing value dsb.
5. *Data Ownership and Distribution*, yaitu siapa yang memiliki data dan bagaimana distribusinya.
6. *Privacy Preservation*, yaitu menjaga kerahasiaan data yang banyak diterapkan pada data nasabah perbankan.
7. *Streaming Data*, yaitu aliran data itu sendiri. (Fajar Astuti Hermawati; 2013: 19).

#### **II.2.6. Manfaat Data Mining**

Data Mining menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan, diantaranya:

1. Membantu menyusun strategi baru untuk meningkatkan sesuatu penjualan.
2. Memberikan Informasi yang berharga, sehingga mampu memberikan kemudahan bagi para pengambil keputusan atau *Decision Maker*.
3. Meningkatkan akses informasi dari mana saja dan kapan saja secara cepat dan mudah.
4. Sangat membantu dalam hal menganalisis informasi.
5. Mampu membaca informasi yang berharga dalam jumlah yang sangat besar.
6. Membantu pemilik usaha untuk menemukan keterkaitan atau pola kemunculan barang dalam transaksi penjualan.

7. Berguna untuk membuat keputusan yang kritis, terutama dalam strategis.
8. Dengan adanya data mining kita dapat menemukan informasi yang berharga dalam tumpukan data yang banyak. (Jurnal SAINTIKOM;Dian Wirdasari,dkk:2011:139-140).

### **II.2.7. Kelemahan Data Mining**

Selain manfaat, data mining juga memiliki beberapa kelemahan, diantaranya:

1. Membutuhkan ketelitian dan kecermatan dalam menerapkan data mining ke aplikasi penggunaannya.
2. Memerlukan waktu yang banyak dalam mengintegrasikan databasenya ke dalam sistem penerapan data miningnya. (Jurnal SAINTIKOM;Dian Wirdasari,dkk:2011:140).

## **II.3. Metode *Association Rules***

### **II.3.1. Pengertian**

Banyak perusahaan yang mengakumulasi sejumlah data yang besar dari transaksi harinya. Kita mulai dengan mengingat masalah perhitungan *item* yang *co-occurence*, yang didasari oleh masalah-masalah seperti analisa keranjang belanja. Suatu keranjang belanja (market basket) merupakan kumpulan item yang dibeli oleh pelanggan dalam satu transaksi pelanggan tunggal. Suatu transaksi pelanggan berisi satu kunjungan tunggal ke sebuah toko, satu pemesanan melalui katalog *mail-order* atau sebuah pemesanan pada sebuah toko di WEB. Tujuan

umum penjual adalah untuk mengidentifikasi item yang dibeli bersama-sama. Informasi ini dapat digunakan untuk meningkatkan susunan barang-barang dalam sebuah toko atau susunan halaman katalog. (Fajar Astuti Hermawati; 2013: 96).

**Tabel II.1. Tabel Pembelian**

<b>Id_Trans</b>	<b>Id_Cust</b>	<b>Tanggal</b>	<b>Item</b>	<b>Jumlah</b>
111	201	5/1/2007	Pena	2
111	201	5/1/2007	Tinta	1
111	201	5/1/2007	Susu	3
111	201	5/1/2007	Jus	6
112	105	6/3/2007	Pena	1
112	105	6/3/2007	Tinta	1
112	105	6/3/2007	Susu	1
113	106	5/10/2007	Pena	1
113	106	5/10/2007	Susu	1
114	201	6/1/2007	Pena	2
114	201	6/1/2007	Tinta	2
114	201	6/1/2007	Jus	4
114	201	6/1/2007	Air	1

(Sumber : Fajar Astuti Hermawati; 2013: 96).

Algoritma asosiasi merupakan suatu bentuk algoritma dalam data mining yang memberikan informasi hubungan antar item data di database. Algoritma tersebut dapat dimanfaatkan secara luas dalam proses bisnis misalnya dalam

proses penjualan. Data mining memberikan hubungan antar data penjualan yang dilakukan pelanggan sehingga akan didapat pola pembelian pelanggan, sehingga pebisnis akan memanfaatkan informasi tersebut untuk mengambil tindakan bisnis yang sesuai. (Jurnal DASI;Emha Taufik Lutfi; 2009).

*Association rules* (aturan asosiasi) atau *affinity analysis* (analisis afinitas) berkenaan dengan studi tentang “apa bersama apa” sebagai contoh dapat berupa studi transaksi di supermarket, misalnya seseorang yang membeli susu bayi susu bayi juga membeli sabun mandi. Pada kasus ini berarti susu bayi bersama dengan sabun mandi. Karena awalnya berasal dari studi tentang database transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan market basket analysis (Santoso, 2007). (Jurnal SAINTIKOM;Dian Wirdasari,dkk:2011: 141).

### **II.3.2. Kaidah Association Rule**

Kaidah Asosiasi atau *Association rules* merupakan mendeteksi kumpulan atribut – atribut yang muncul bersamaan (co-occur) dalam frekuensi yang sering, dan membentuk sejumlah kaidah dari kumpulan – kumpulan tersebut.

Association rule merupakan sebuah ekspresi implikasi yang berbentuk  $X \rightarrow Y$ , dimana X dan Y merupakan disjoint itemset  $(X \cap Y) = \emptyset$ . Contoh : {Pena, Tinta} @ {Jus}. Dalam *association rule*, kita dapat menghitung support dan *confidence*. *Confidence* menyatakan seberapa sering item-item Y muncul dalam transaksi yang berisi X. Secara formal dapat dinyatakan dengan persamaan berikut ini : (Fajar Astuti Hermawati; 2013: 99).

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Dimana  $s$  adalah *Support* dan  $c$  adalah *Confidence* .

Contoh : Untuk *association rule* {Pena,Tinta}@{Jus}, maka :

$$s(X \rightarrow Y) = \frac{\sigma(\{Pena, Tinta, Jus\})}{4} = \frac{2}{4} = 0.5$$

$$c(X \rightarrow Y) = \frac{\sigma(\{Pena, Tinta, Jus\})}{\sigma(\{Pena, Tinta\})} = \frac{2}{3} = 0.67$$

(Sumber : Fajar Astuti Hermawati; 2013: 99).

### II.3.3. Association Rule Mining

Jika terdapat sebuah himpunan transaksi  $T$ , maka tujuan dari *association rule mining* adalah untuk menemukan semua aturan yang mempunyai *support*  $\geq$  *minsup* dan *confidence*  $\geq$  *minconf* . Pendekatan *brute-force* untuk *association rule mining* menggunakan pendekatan dengan menghitung *support* dan *confidence* dari semua kemungkinan rule. Pendekatan *brute-force* terdiri dari langkah-langkah berikut ini : (Sumber : Fajar Astuti Hermawati; 2013: 99-100).

- a. Daftar semua kemungkinan *association rules*.
- b. Hitung *support* dan *confidence* untuk setiap rule.
- c. Pangkas rule yang tidak memenuhi *minsup* dan *minconfthresholds*.

Secara umum, pendekatan ini sangat mahal dalam waktu komputasinya. Jumlah kemungkinan rule dari sebuah tabel transaksi yang terdiri dari  $d$  item adalah sebesar :

$$\mathbf{R = 3^d - 2^{d+1} + 1}$$

(Sumber : Fajar Astuti Hermawati; 2013:100).

Misalkan digunakan data pada tabel pembelian di atas dengan jumlah item  $d = 5$ , maka jumlah kemungkinan rule adalah sebesar :

$$\mathbf{R = 3^5 - 2^{5+1} + 1 = 243 - 64 + 1 = 180}$$

(Sumber : Fajar Astuti Hermawati; 2013:100).

Contoh : Jika diketahui data transaksi sebagai berikut ;

**Tabel II.2. Tabel Transaksi**

<b>Id_Trans</b>	<b>Items</b>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

(Sumber : Fajar Astuti Hermawati; 2013:100)

Maka support dan confidence dari association rule sebagai berikut :

{Milk, Diaper}@{Beer} (s = 0.4, c = 0.67)

{Milk, Beer}@{Diaper} (s = 0.4, c = 1.0)

{Diaper, Beer}@{Milk} (s = 0.4, c = 0.67)

{Beer}@{Milk, Diaper} (s = 0.4, c = 0.67)

$\{\text{Diaper}\} \otimes \{\text{Milk, Beer}\} (s = 0.4, c = 0.5)$

$\{\text{Milk}\} \otimes \{\text{Diaper, Beer}\} (s = 0.4, c = 0.5)$

Semua rule di atas merupakan partisi biner dari itemset yang sama : {Milk, Diaper, Beer}. Rule yang berasal dari *itemset* yang sama mempunyai *support* yang identik tetapi mempunyai *confidence* yang berbeda, sehingga kita dapat memasangkan ulang untuk keperluan *support* dan *confidence*. (Sumber : Fajar Astuti Hermawati; 2013:100)

Dari observasi dengan menggunakan contoh di atas maka dapat digunakan pendekatan dua langkah yaitu :

1. FrequentItemset Generation

Yaitu membangkitkan semua itemset  $s$  yang  $\text{support}^3$  minsup.

2. Rule Generation

Yaitu membangkitkan rule dengan *confidence* tinggi dari setiap frequentitemset, dimana setiap rule merupakan partisi biner dari sebuah frequentitemset.

## II.4. Database atau Basis Data

*Database* atau basis data adalah suatu susunan / kumpulan data operasional lengkap dari suatu organisasi / perusahaan yang diorganisir / dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan computer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Sistem Database atau sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Berikut ini beberapa istilah yang dipergunakan di dalam sistem basis data:

- a. Enterprise, suatu bentuk organisasi seperti bank, universitas, pabrik dan lain-lain.
- b. Entitas, suatu objek yang dapat dibedakan dengan objek lainnya yang dapat diwujudkan di dalam basis data.
- c. Attribute/Field, karakteristik entitas tertentu.
- d. Data value (nilai atau isi data) merupakan data *actual* atau informasi yang disimpan di tiap data elemen atau atribut. Isi atribut disebut nilai data.
- e. Record/tuple, kumpulan isi elemen data (atribut) yang saling berhubungan menginformasikan tentang suatu entity secara lengkap.
- f. File, kumpulan record sejenis yang mempunyai panjang elemen dan atribut yang sama, namun berbeda-beda data *value*-nya.
- g. Kunci elemen data, sebagai tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.
- h. Database Management System (DBMS), merupakan kumpulan software yang mengkoordinasikan semua kegiatan yang berhubungan dengan basis data agar data dapat diakses/dipakai oleh pemakai / *user*. Tujuannya adalah efisiensi dan

kenyamanan dalam memperoleh dan menyimpan informasi di dalam basis data. (Aryo Nugroho; 2009: 11)

## II.5. Microsoft Visual Studio.Net 2010

Pemrograman Microsoft Visual Studio .NET 2010 adalah sebuah *platform* untuk membangun, menjalankan dan meningkatkan generasi lanjut dari aplikasi terdistribusi.

Net *platform* merupakan *platform* terbaru untuk pemrograman aplikasi window dari microsoft dalam upaya meningkatkan produktivitas pembuatan sebuah program aplikasi dan memungkinkan terbukanya peluang untuk menjalankan program pada *multi sistem operasi* serta dapat memperluas pengembangan aplikasi *Client-Server*.



**Gambar II.1. Visual Studio 2010**

(Sumber : [www.google.com](http://www.google.com))

## II.6. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS ( *Relational Database Management System* ) yang sangat powerful dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbaru ini, SQL Server 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa database. (Aryo Nugroho; 2009:1).

SQL Server 2008 memiliki suatu GUI ( *Graphic User Interface* ) yang dapat digunakan untuk melakukan aktivitas sehari-hari berkaitan dengan database, seperti menulis T-SQL, melakukan backup dan restore database, melakukan security database terhadap aplikasi dan sebagainya. Pada GUI tersebut terdapat setingan terhadap SQL Server untuk bekerja lebih optimal. Settingannya juga bisa dilakukan dengan menggunakan script untuk memudahkan developer mengubah setting pada SQL Server 2008. (Aryo Nugroho; 2009 : 1).

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

UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemrograman yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object software*

*Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu: identifikasi kelas – kelas dan obyek – obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design objek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung OO. Metode OOSE dari Jacobson lebih memberi penekanan pada *Use Case*, OOSE memberikan tiga tahap yaitu membuat model requirement dan analisis, design dan implementasi, dan model – model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML Metode Booch, OMT, dan OOSE digabungkan dengan membuang elemen – elemen yang tidak praktis ditambah dengan elemen – elemen dari metode lain yang lebih efektif dan elemen – elemen baru yang belum ada pada metode terdahulu sehingga lebih ekspresif dan seragam daripada metode lainnya. Gambar berikut ini adalah unsur – unsur yang membentuk UML. (Munawar;2005:17-18)



**Gambar II.2. Unsur – unsur pembentuk UML**

(Sumber : Munawar 2005 ; 18)

### II.7.1. Tujuan Pemanfaatan UML

Berikut tujuan utama dalam desain UML adalah :

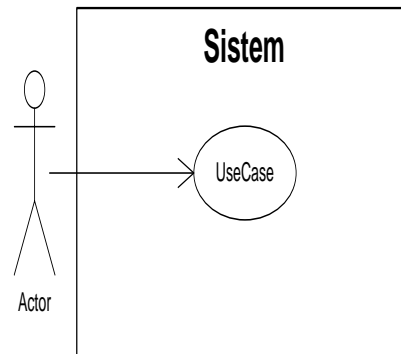
- a. Memberikan model yang siap pakai bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dimengerti secara umum.
- b. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- c. Menyatukan praktek – praktek yang terbaik yang terdapat dalam pemodelan.

### II.7.2. Struktur Diagram dalam *Unified Modeling Language* (UML)

Ada 5 (lima) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

a. *Use Case Diagram*

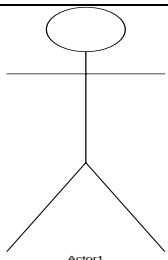

Diagram use case menunjukkan 3 aspek dari system yaitu actor, use case dan system/sub system boundary. Actor mewakili peran orang, system yang lain atau ketika berkomunikasi dengan use case. (Munawar;2005:64)



**Gambar II.3: use case diagram**

**(Sumber : Munawar 2005 : 64)**

Berikut ini keterangan dari simbol – simbol yang ada dalam use case diagram :

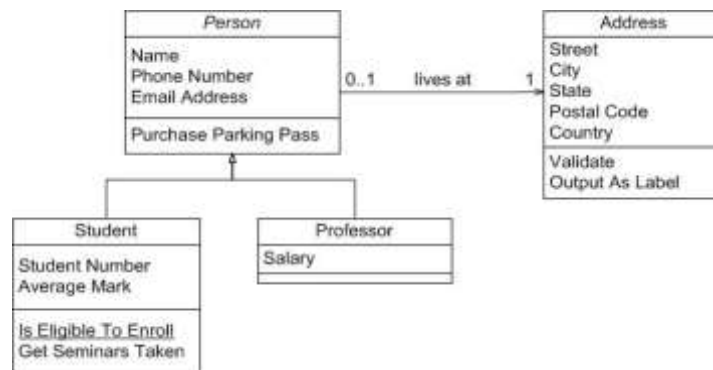
Simbol	Keterangan
	Actor, adalah abstraction dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem
	Usecase, adalah abstraksi dari interaksi antara sistem dan aktor

**Tabel II.3. Keterangan Simbol – Simbol Use Case**

**(Sumber : Munawar 2005 : 64-65)**

### b. *Class Diagram*

Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena class adalah deskripsi kelompok obyek – obyek dengan property, perilaku (operasi).(Munawar;2005:219).



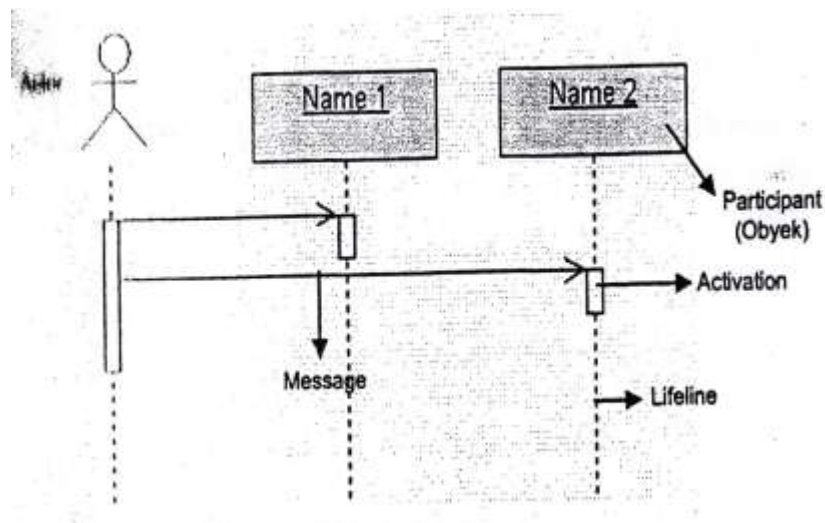
**Gambar II.4 : *Class Diagram***

**(Sumber : Munawar : 2005 ; 220)**

### c. *Sequence Diagram*

Menurut Munawar (2005:87) Diagram ini digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesen) yang diletakkan diantara obyek-obyek use case. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

Berikut adalah Simbol-simbol yang ada pada *Sequence Diagram* yaitu:



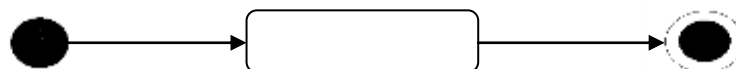
**Gambar II.5. Simbol –simbol yang ada pada *Sequence Diagram***

(Sumber: Munawar,2005 : 89)

*d. State Chart Diagram*

Menurut Munawar ( 2005:73 ). State diagram menyediakan variasi symbol dan sejumlah ide untuk pemodelan. Simbol UML untuk State transition diagram adalah segi empat yang tiap pojoknya dibuat *rounded*. Titik awalnya menggunakan lingkaran solid yang diarsir dan diakhiri dengan mata.

Berikut adalah simbol UML untuk *State Chart Diagram*:





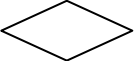


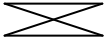
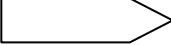
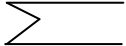

**Gambar II.6. *State Chart Diagram***

(Sumber: Munawar, 2005 : 74)

e. *Activity Diagram*

Menurut Munawar (2005:109) Activity diagram adalah teknik mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam kasus, activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak. Berikut simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram* yaitu:

**Tabel II.4. Simbol Activity Diagram**

Notasi	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Rake</i> menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir ( <i>Flow Final</i> )

(Sumber :Munawar, 2005:110)

## II.8. Entity Relationship Diagram

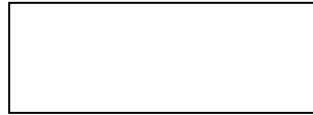
*Entity Relationship Model/ER\_M* merupakan suatu model data yang dikembangkan berdasarkan obyek. ERM digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logic. ERM didasarkan pada suatu persepsi bahwa *real world* terdiri atas obyek – obyek dasar yang mempunyai hubungan atau kerelasian antar obyek – obyek dasar tersebut (Martin, 1975). ERM digambarkan dalam bentuk diagram ER. Untuk menggambarkan ERM digunakan simbol-simbol grafis tertentu. (Edhy Sutanta; 2011:91).

Perancangan basis data dengan menggunakan model *entity relationship* adalah dengan menggunakan *Entity Relationship Diagram* (ERD). Terdapat tiga notasi dasar yang bekerja pada model E-R yaitu : ***entity sets, relationship sets, dan attributes***. (Kusrini; 2007:21)

Sebuah entity adalah sebuah “benda” (thing) atau “objek” (object) di dunia nyata yang dapat dibedakan dari semua objek lainnya. Entity sets adalah sekumpulan entity yang mempunyai tipe yang sama. Kesamaan tipe ini dapat dilihat dari atribut/property yang dimiliki oleh setiap entity. Misal :

1. Kumpulan orang yang menyimpan uang pada suatu bank dapat didefinisikan sebagai entity set nasabah.
2. Kumpulan orang yang belajar di perguruan tinggi didefinisikan sebagai mahasiswa.

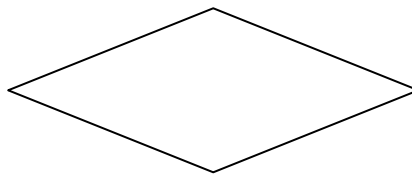
Entity set dilambangkan dengan bentuk persegi panjang, seperti tampak pada gambar berikut :



**Gambar II.7. Lambang Entity Set**

**(Sumber : Kusrini; 2007:21)**

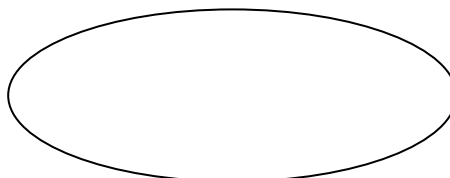
Relationship adalah hubungan diantara beberapa entiti. Relationship set adalah sekumpulan relasi yang mempunyai tipe yang sama. Relationship set digambarkan dengan diamond seperti tampak pada gambar di bawah ini.



**Gambar II.8. Lambang Relationship Set**

**(Sumber : Kusrini; 2007:21)**

Atribut merupakan sebutan untuk mewakili suatu entiti. Atribut dalam ERD dilambangkan dengan bentuk elips, seperti tampak pada gambar di bawah ini.



**Gambar II.9. Lambang Atribut**

**(Sumber : Kusrini; 2007:22)**

## II.9. Normalisasi (*Normalization*)

### II.9.1. Definisi Normalisasi

Perancangan basis data menghasilkan sekumpulan relasi yang saling berkerelasian dalam lingkup sebuah sistem. Untuk memenuhi batasan dalam definisi basis data maka setiap rancangan relasi perlu diuji untuk menentukan apakah relasi tersebut telah optimal. Pengujian dilakukan berdasarkan kriteria tertentu. Jika relasi belum optimal maka perlu dilakukan proses normalisasi. Perjudan normalisasi adalah dekomposisi relasi menjadi relasi – relasi baru yang lebih sederhana. (Edhy Sutanta; 2011:174).

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan / mendekomposisi data dalam 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 in-efisiensi pengolahan (Martin, 1975). (Edhy Sutanta; 2011:174).

Salah satu topik yang cukup kompleks dalam dunia manajemen database adalah proses untuk menormalisasi tabel-tabel dalam database relasional. Hal ini akan membahas mengenai bentuk-bentuk normalisasi dan teknik-teknik yang digunakan dalam melakukan normalisasi. (Kusrini; 2007:39)

Dengan normalisasi kita ingin mendesain database relasional yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.

3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Mendhindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi database minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada database yang tidak normal. (Kusrini; 2007:39-40).

“Insertion anomaly” adalah sebuah kesalahan dalam penempatan informasi entri data baru ke seluruh tempat dalam database dimana informasi tersebut perlu disimpan. Dalam database yang telah dinormalisasi, proses pemasukan suatu informasi baru hanya perlu dimasukkan ke dalam suatu tempat.

“Deletion anomaly” adalah sebuah kesalahan dalam penghapusan suatu informasi dalam database harus dilakukan dengan penghapusan informasi dalam database harus dilakukan dengan penghapusan informasi tersebut dari beberapa tempat dalam database. Dalam database yang telah dinormalisasi, penghapusan suatu informasi hanya perlu dilakukan dalam satu tempat dalam database tersebut.

Sedangkan melakukan update satu informasi, kesalahan yang dapat terjadi ketika kita terus melakukan update ke seluruh tempat yang menyimpan informasi tersebut. Kesalahan ini disebut dengan “update anomaly”. (Kusrini; 2007:39-40).

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. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari model E-R ke bentuk fisik. (Kusrini; 2007:39-40).

## **II.9.2. Bentuk – Bentuk Normalisasi**

### **1. Bentuk Tidak Normal**

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti formal tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya. (Kusrini; 2007:39-41).

### **2. Bentuk Normal Tahap Pertama (1<sup>st</sup> Normal Form)**

Relasi ini disebut juga sebagai 1NF jika memenuhi kriteria sebagai berikut :

- a. Jika seluruh atribut dalam relasi bernilai atomic (atomic value).
- b. Jika seluruh atribut dalam relasi bernilai tunggal (single value).
- c. Jika relasi tidak memuat set atribut berulang.
- d. Jika semua record mempunyai sejumlah atribut yang sama.

Permasalahan dalam 1NF adalah sebagai berikut :

- a. Tidak dapat menyisipkan informasi parsial.
- b. Terhapusnya informasi ketika menghapus sebuah record.
- c. Pembaruan atribut nonkunci mengakibatkan sejumlah record harus diperbaharui.(Edhy Sutanta; 2011:176).

Mengubah bentuk relasi UNF menjadi bentuk 1NF dapat dilakukan dengan cara sebagai berikut :

- a. Melengkapi nilai-nilai dalam atribut.
- b. Mengubah struktur relasi.

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

Relasi disebut juga sebagai 2NF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 1NF.
- b. Jika semua atribut nonkunci FD pada PK.

Permasalahan dalam 2NF adalah sebagai berikut:

- a. Kerangkapan data (data redundancy).
- b. Pembaruan yang tidak benar dapat menimbulkan inkonsistensi data (data inconsistency).
- c. Proses pembaharuan data tidak efisien.
- d. Penyimpangan pada saat penyisipan, penghapusan, dan pembaruan.

Kriteria tersebut mengindikasikan bahwa di antara atribut dalam 2NF masih mungkin mengalami TDF. Selain itu, relasi 2NF menuntut telah didefinisikan atribut PK dalam relasi. Mengubah relasi 1NF menjadi bentuk 2NF dapat dilakukan dengan mengubah struktur relasi dengan cara :

- a. Identifikasikan FD relasi 1NF (jika perlu gambarkan diagram ketergantungan datanya).
- b. Berdasarkan informasi tersebut, dekomposisi relasi 1NF menjadi relasi-relasi baru sesuai FD-nya. Jika menggunakan diagram maka simpul-simpul yang berada pada puncak diagram ketergantungan data bertindak sebagai PK pada relasi baru. (Edhy Sutanta; 2011:177).

#### 4. Bentuk Normal Ketiga (3<sup>rd</sup> normal form/3NF)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \geq A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- a. X haruslah superkey pada tabel tersebut
- b. A merupakan bagian dari primary key pada tabel tersebut. (Kusrini; 2007:39-42).

#### 5. Bentuk Normal Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk normal keempat berkaitan dengan sifat ketergantungan banyak nilai (Multivalued Dependency) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari Project Join Normal Form (PJNF). (Kusrini; 2007:39-43).

#### 6. Bentuk Normal Boyce-Codd (Boyce-Codd Norm Form/BCNF)

Bentuk normal BCNF dikemukakan oleh R.F. Boyce dan E.F. Codd. Suatu relasi disebut sebagai BCNF jika memenuhi kriteria sebagai berikut :

- a. Jika memenuhi kriteria 3NF.
- b. Jika semua atribut penentu (determinan) merupakan *constraint key*.