

## BAB II

### LANDASAN TEORI

#### 2.1 *Perancangan*

Analisis dan perancangan sistem seperti yang ditampilkan oleh penganalisis sistem berupaya menganalisis *input* data atau aliran data secara sistematis, memproses atau mentransformasikan data, menyimpan data dan menghasilkan output informasi dalam konteks bisnis khusus. (Kenneth E. Kendall dan Julie E. Kendall, 2010:7)

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut :

1. *Identify* yaitu mengidentifikasi masalah.

Mengidentifikasi (mengetahui) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat diidentifikasi sebagai suatu pernyataan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai.

2. *Understand* yaitu memahami cara kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara rinci bagaimana sistem yang ada beroperasi.

3. *Analyze* yaitu menganalisis sistem.

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan. Menganalisis hasil penelitian sering kali dilakukan oleh analis sistem yang masih baru.

## 2.2 Sistem

Sistem berasal dari bahasa Yunani yang artinya kesatuan. Di dalam mendefinisikan pengertian sistem ada dua kelompok pendekatan yang harus diperhatikan yaitu menekankan pada prosedur dan menekankan pada komponen atau elemennya.

Berikut pengertian sistem menurut para ahli :

1. Menurut Kenneth E. Kendall dan Julie E. Kendall (2010:523), “suatu sistem adalah serangkaian subsistem yang saling terkait dan tergantung satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah ditetapkan sebelumnya. Semua sistem memiliki *input*, proses, *output*, dan umpan balik.”

Dari definisi di atas dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu:

1. Setiap sistem terdiri lebih dari satu komponen.
2. Komponen-komponen dalam sistem saling berinteraksi dan terhubung.
3. Suatu sistem mempunyai pola yang terpadu untuk melaksanakan kegiatan.

Berdasarkan definisi di atas, sistem dapat diartikan sebagai kumpulan dari beberapa komponen yang saling terhubung dan bekerjasama untuk mencapai

tujuan. Sistem ini berjalan berdasarkan pola yang telah dibuat dengan aturan-aturan operasi sistem yang terdiri dari masukan, proses, keluaran, serta suatu totalis yang digerakkan oleh suatu subsistem yang tidak kaitannya dengan sistem yang lebih luas.

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, seperti:

### 1. Komponen-komponen (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen.

### 2. Batas sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

### 3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan

luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

#### 4. Penghubung (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu sistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung.

#### 5. Masukan (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapat keluaran.

#### 6. Keluaran (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

#### 7. Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

## 8. Sasaran (*Objectives*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. (Sutabri, 2008:14)

### 2.3 *Representasi Pengetahuan*

Representasi pengetahuan adalah suatu bentuk teknik untuk merepresentasikan basis pengetahuan yang diperoleh kedalam suatu skema diagram tertentu sehingga dapat diketahui relasi/keterhubungan antara suatu data dengan data lainnya (Arhami, 2010:53)

Beberapa model representasi yang penting adalah :

#### 1. Jaringan Semantik (*Semantic Nets*)

Jaringan semantik adalah teknik representasi pengetahuan yang digunakan untuk informasi proposional, sedangkan yang dimaksudkan dengan informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah.

#### 2. Bingkai (*Frame*)

Bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi ataupun elemen-elemen lainnya.

### 3. Logika Predikat (*PredicateLogic*)

Logika predikat adalah logika yang digunakan untuk menganalisis struktur *internal* kalimat. Logika predikat ditinjau berdasarkan pada kebenaran dan kaidah inferensi untuk merepresentasikan simbol dan hubungan satu dengan yang lain. Logika predikat digunakan untuk menentukan kebenaran atau kesalahan sebuah pernyataan, selain itu logika predikat digunakan untuk merepresentasikan pernyataan tentang objek tertentu.

### 4. Kaidah Produksi (*ProductionRule*)

Cara merepresentasikan pengetahuan berbasis kaidah adalah dengan menggunakan pernyataan *if – then*. Di mana bagian *then* akan bernilai benar jika satu atau lebih sekumpulan fakta atau hubungan antar fakta diketahui benar, memenuhi bagian *if*. Secara umum, dalam bentuk kaidah produksi *IF* premis *Then* konklusi; maka untuk premis yang lebih dari satu dapat dihubungkan dengan operator *and* atau *or*. Sedangkan bagian konklusi dapat berupa kalimat tunggal, beberapa kalimat yang dihubungkan dengan *and* dan dimungkinkan dikembangkan dengan *else* (Sri Kusumadewi, 2009:22).

Kecepatan dan ketepatan sistem pakar melakukan penalaran untuk memberikan keluaran banyak tergantung pada kaidah - kaidah yang disimpan sebagai basis pengetahuan dan mekanisme inferensi yang memiliki kaidah yang tepat untuk menghasilkan

keluaran. Dalam pembangunan sistem berbasis pengetahuan, pengetahuan yang telah diekstrak dimasukkan ke dalam program komputer oleh proses yang disebut representasi pengetahuan (*knowledge representation*). Sistem representasi pengetahuan merupakan gabungan dari dua elemen yakni struktur data dan prosedur menafsirkan (*interpretive procedure*) untuk pemakaian pengetahuan yang dimasukkan dalam struktur data (Sri Kusumadewi, 2009:68).

#### **2.4 *Machine Learning***

*Machine learning* adalah cabang dari kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data. Sistem pembelajaran dapat memanfaatkan contoh (data) untuk menangkap ciri yang diperlukan dari probabilitas yang mendasari (yang tidak diketahui). Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara *variable* yang diamati. Fokus besar penelitian pembelajaran mesin adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua perilaku yang mungkin, dari semua masukan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan (data pelatihan). Karena itu pembelajaran harus merampatkan (generalisasi) perilaku

dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru.

Mesin disini adalah mesin dalam pengertian lebih mendekati ‘*sistem*’ bukan mesin mekanik. Istilah pembelajaran pertama kali muncul pada disiplin ilmu kecerdasan buatan. Pembelajaran berarti menambah pengetahuan, memahami dengan belajar, mengikuti perintah. Pembelajaran mesin merupakan salah satu cabang dari kecerdasan buatan yang membahas mengenai pembangunan sistem yang didapat berdasarkan pada pembelajaran data, atau sebuah studi yang mempelajari cara untuk memprogram sebuah komputer untuk belajar. Inti dari pembelajaran mesin adalah representasi dan generalisasi. Pada tahun 1959, Arthur Samuel mendefinisikan bahwa pembelajaran mesin adalah bidang studi yang memberikan kemampuan untuk belajar tanpa deprogram secara eksplisit. Kemampuan belajar yang menjadi dominan ditentukan oleh kemampuan perangkat lunak atau algoritmanya. Implementasi kemampuan belajar dapat dicapai dengan berbagai teknik, ada yang menggunakan kaidah(*rule*), ada yang menggunakan statistika, ada yang menggunakan pendekatan fisiologi yaitu sistem saraf manusia atau disebut dengan ANN(*ArtificialNeuralNetwork*) atau jaringan saraf tiruan. Pembelajaran mesin dapat berfungsi untuk beradaptasi dengan suatu keadaan yang baru, serta untuk mendeteksi dan memperkirakan suatu pola.

## **2.5    *Metode Bayes***

### **2.5.1   *TeoremaBayes***

*Bayes* merupakan teknik prediksi berbasis probabilistik sederhana yang berdasarkan pada penerapan *teorema Bayes* (atau aturan *Bayes*) dengan asumsi independensi (ketidaktergantungan) yang kuat (naif). Dengan kata lain, *Naïve Bayes*, model yang digunakan adalah “model fitur independen”.

Dalam *Bayes* (terutama *Naïve Bayes*), maksud independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama.

Prediksi *Bayes* didasarkan pada *teorema Bayes* dengan formula umum sebagai berikut :

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

Penjelasan dari formula tersebut adalah sebagai berikut :

Parameter	Keterangan
P(H E)	Probabilitas akhir bersyarat ( <i>conditional Probability</i> ) suatu hipotesis H terjadi jika diberikan bukti ( <i>evidence</i> ) E terjadi.
P(E H)	Probabilitas sebuah bukti E terjadi akan mempengaruhi hipotesis H.
P(H)	Probabilitas awal (priori) hipotesis H terjadi tanpa memandang bukti apapun
P(E)	Probabilitas awal (priori) bukti E terjadi tanpa memandang hipotesis/bukti yang lain

Ide dasar dari aturan *Bayes* adalah bahwa hasil dari hipotesis atau peristiwa (H) dapat diperkirakan berdasarkan pada beberapa bukti (E) yang diamati. Ada beberapa hal penting dari aturan *Bayes* tersebut, yaitu :

1. Sebuah Probabilitas awal/prior  $H$  atau  $P(H)$  adalah probabilitas dari suatu hipotesis sebelum bukti diamati.

2. Sebuah probabilitas akhir  $H$  atau  $P(H|E)$  adalah probabilitas dari suatu hipotesis setelah bukti diamati.

### 2.5.2 Karakteristik *NaïveBayes*

Klasifikasi dengan *NaïveBayes* bekerja berdasarkan teori probabilitas yang memandang semua fitur dari data sebagai bukti dalam probabilitas. Hal ini memberikan karakteristik *NaïveBayes* sebagai berikut :

*NaïveBayes* bekerja teguh (*robust*) terhadap data-data yang terisolasi yang biasanya merupakan data dengan karakteristik berbeda (*outliner*). *NaïveBayes* juga bisa menangani nilai atribut yang salah dengan mengabaikan data latih selama proses pembangunan model dan prediksi.

1. Tangguh menghadapi atribut yang tidak relevan.
2. Atribut yang mempunyai korelasi bisa mendegradasi kinerja klasifikasi *NaïveBayes* karena asumsi independensi atribut tersebut sudah tidak ada.

## 2.6 *Klasifikasi*

### 2.6.1 Konsep Klasifikasi

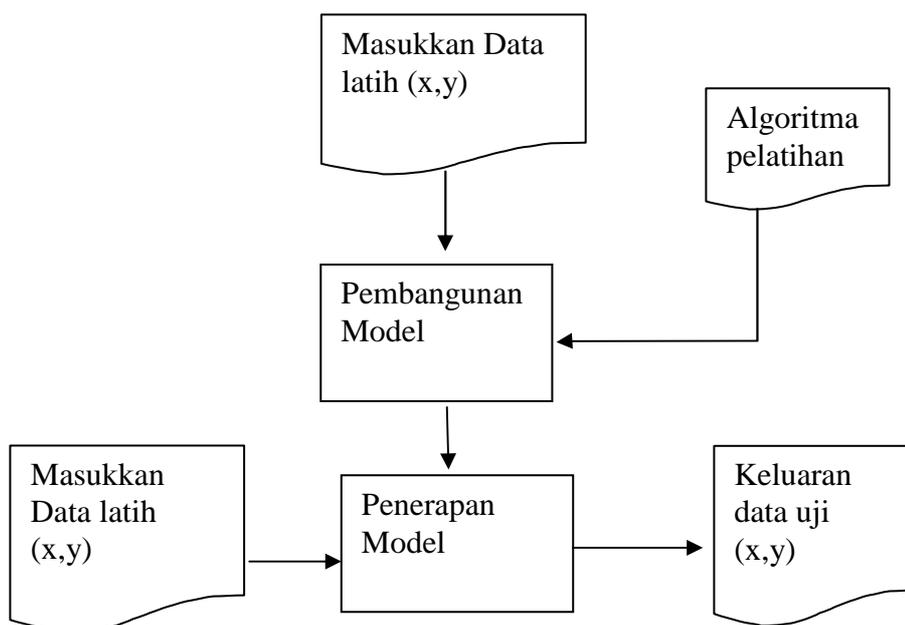
Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkan ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu : pertama, pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan kedua, penggunaan model tersebut untuk melakukan

pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang mudah disimpan.

Contoh aplikasi yang sering ditemui adalah pengklasifikasian jenis hewan, yang mempunyai sejumlah atribut. Dengan atribut tersebut, jika ada hewan baru, kelas hewannya bisa langsung diketahui. Contoh lain adalah bagaimana melakukan diagnosis penyakit kulit kanker melanoma (Amaliyah et al, 2011), yaitu dengan melakukan pembangunan model berdasarkan data latih yang ada, kemudian menggunakan model tersebut untuk mengidentifikasi penyakit pasien baru sehingga diketahui apakah pasien tersebut menderita kanker atau tidak.

### **2.6.2 Model Klasifikasi**

Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. Kerangka kerja (*framework*) klasifikasi ditunjukkan pada gambar. Pada gambar tersebut disediakan sejumlah data latih  $(x,y)$  untuk digunakan sebagai data pembangunan model. Model tersebut kemudian dipakai untuk memprediksi kelas dari data uji  $(x,y)$  sehingga diketahui kelas  $y$  yang sesungguhnya.



Gambar 2.2 Flowchart Model Klasifikasi.

Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label kelas baru yang belum diketahui. Dalam pembangunan model selama proses pelatihan tersebut diperlukan suatu algoritma untuk membangunnya, yang disebut algoritma pelatihan (*learning algorithm*). Ada banyak algoritma pelatihan yang sudah dikembangkan oleh para peneliti, seperti *K-Nearest Neighbor*, *Artificial Neural Network*, *Support Vector Machine* dan sebagainya. Setiap algoritma mempunyai kelebihan dan kekurangan, tetapi semua algoritma berprinsip sama, yaitu melakukan suatu pelatihan sehingga di akhir pelatihan, model dapat memetakan (memprediksi) setiap *vektor* masukan ke label kelas keluaran dengan benar.

### 2.6.3 Pengukuran Kinerja Klasifikasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar, tetapi tidak dapat dipungkiri bahwa kinerja suatu sistem tidak bisa 100% benar sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya. Umumnya, pengukuran kinerja klasifikasi dilakukan dengan matriks konfusi (*confusion matrix*).

Matriks konfusi merupakan table pencatat hasil kerja klasifikasi. Kuantitas matriks konfusi dapat diringkus menjadi dua nilai, yaitu akurasi dan laju *error*. Dengan mengetahui jumlah data yang diklasifikasikan secara benar, kita dapat mengetahui akurasi hasil prediksi dan dengan mengetahui jumlah data yang diklasifikasikan secara salah, kita dapat mengetahui laju *error* dari prediksi yang dilakukan. Dua kuantitas ini digunakan sebagai matrik kinerja klasifikasi. Untuk menghitung akurasi digunakan formula.

$$\text{akurasi} = \frac{\text{jumlah data yang diprediksi secara benar}}{\text{jumlah prediksi yang dilakukan}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \dots \dots \dots$$

Untuk menghitung laju *error* (kesalahan prediksi) digunakan formula

$$\text{laju error} = \frac{\text{jumlah data yang diprediksi secara salah}}{\text{jumlah prediksi yang dilakukan}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \dots \dots \dots$$

Semua algoritma klasifikasi berusaha membentuk model yang mempunyai akurasi tinggi atau (laju *error* yang rendah). Umumnya, model yang dibangun memprediksikan dengan benar pada semua data yang menjadi data latihnya, tetapi

ketika model berhadapan dengan data uji, barulah kinerja model dari sebuah algoritma klasifikasi ditentukan.

## 2.7 *Naïve Bayes Untuk Klasifikasi*

Kaitan antara *Naïve Bayes* dengan klasifikasi, korelasi hipotesis dan bukti klasifikasi adalah bahwa hipotesis dalam *teorema Bayes* merupakan label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan bukti merupakan fitur-fitur yang menjadikan masukkan dalam model klasifikasi. Jika  $X$  adalah *vector* masukkan yang berisi fitur dan  $Y$  adalah label kelas, *Naïve Bayes* dituliskan dengan  $P(X|Y)$ . Notasi tersebut berarti probabilitas label kelas  $Y$  didapatkan setelah fitur-fitur  $X$  diamati. Notasi ini disebut juga probabilitas akhir (*posterior probability*) untuk  $Y$ , sedangkan  $P(Y)$  disebut probabilitas awal (*prior probability*)  $Y$ .

Selama proses pelatihan harus dilakukan pembelajaran probabilitas akhir  $P(Y|X)$  pada model untuk setiap kombinasi  $X$  dan  $Y$  berdasarkan informasi yang didapat dari data latih. Dengan membangun model tersebut, suatu data uji  $X'$  dapat diklasifikasikan dengan mencari nilai  $Y'$  dengan memaksimalkan nilai  $P(X'|Y')$  yang didapat.

Formulasi *Naïve Bayes* untuk klasifikasi adalah :

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)}$$

$P(X|Y)$  adalah probabilitas data dengan *vector*  $X$  pada kelas  $Y$ .  $P(Y)$

Adalah probabilitas awal kelas  $Y \cdot \prod_{i=1}^q P(X_i|Y)$  adalah probabilitas independen kelas  $Y$  dari semua fitur dalam *vector*  $X$ . Nilai  $P(X)$  selalu tetap sehingga dalam perhitungan prediksi nantinya kita tinggal menghitung bagian  $P(Y) \prod_{i=1}^q P(X_i|Y)$  dengan memilih yang terbesar sebagai kelas yang dipilih sebagai hasil prediksi. Semua probabilitas independen  $\prod_{i=1}^q P(X_i|Y)$  tersebut merupakan pengaruh semua fitur dari data terhadap setiap kelas  $Y$ , yang di notasikan dengan :

$$P(X|Y=y) = \prod_{i=1}^q P(X_i|Y)$$

Setiap set fitur  $X = \{X_1, X_2, X_3, \dots, X_q\}$  terdiri atas  $q$  atribut ( $q$  dimensi).

Umumnya, *Bayes* mudah dihitung untuk fitur bertipe kategoris seperti pada kasus klasifikasi hewan dengan fitur “penutup kulit dengan nilai {bulu, rambut, cangkang} atau kasus fitur “jenis kelamin” dengan nilai {pria, wanita}. Namun untuk fitur dengan tipe numeric (kontinu) ada perlakuan khusus sebelum dimasukkan dalam *Naïve Bayes*. Caranya adalah :

1. Melakukan diskretisasi pada setiap fitur kontinu dan mengganti nilai fitur kontinu tersebut dengan nilai interval diskret. Pendekatan ini dilakukan dengan mentransformasikan fitur kontinu ke dalam fitur ordinal.
2. Mengasumsikan bentuk tertentu dari distribusi probabilitas untuk fitur kontinu dan memperkirakan parameter distribusi dengan data pelatihan. Distribusi *Gaussian* biasanya dipilih untuk mempresentasikan probabilitas bersyarat dari fitur kontinu pada sebuah kelas  $P(X_i|Y)$ , sedangkan distribusi *Gaussian* dikarakteristikan dengan dua parameter : mean,  $\mu$  dan

varian,  $\sigma^2$ . Untuk setiap kelas  $y_j$ , probabilitas bersyarat kelas  $y_j$  untuk fitur  $X_i$  adalah :

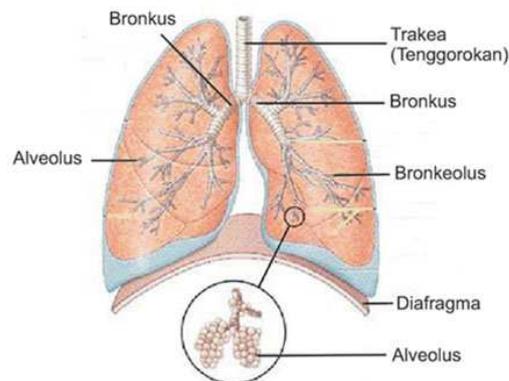
$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \exp \left[ -\frac{(X_i - \mu_{ij})^2}{2\sigma_{ij}^2} \right]$$

Parameter  $\mu_{ij}$  bisa didapat dari mean sampel  $X_i$  ( $\bar{x}$ ) dari semua data latih yang menjadi milik kelas  $y_j$ , sedangkan  $\sigma_{ij}^2$  dapat diperkirakan dari varian sampel ( $s^2$ ) dari data latih.

## 2.8 Paru-paru

Paru-paru adalah organ pada sistem pernapasan (*respirasi*) dan berhubungan dengan sistem peredaran darah (*sirkulasi*) vertebrata yang bernapas dengan udara. Fungsinya adalah menukar oksigen dari udara dengan *karbondioksida* dari darah.

Berikut adalah bagian dari paru-paru dan fungsinya :



Gambar 2.3 Bagian Paru-paru.

1. *Bronkus* adalah batang bercabang yang menghubungkan paru-paru kiri, paru-paru kanan, dan trakea. *Bronkus* tersusun atas tulang rawan, lapisan

*mukosa*, dan otot polos. Tulang rawan berfungsi sebagai rangka *bronkus*, lapisan mukosa menghasilkan lendir untuk menjebak partikel asing yang akan memasuki paru-paru, dan otot polos membuat kita dapat bernapas secara otomatis tanpa disadari.

2. *Bronkiolus* adalah cabang dari *bronkus* yang bermuara ke *alveolus*. Struktur *bronkus* tidak memiliki tulang rawan, memiliki *silia*, dan di bagian ujung terdiri dari jaringan *epitelium* berbentuk kubus bersilia.
3. *Alveolus* adalah tempat melakukan pertukaran antara oksigen dan karbon dioksida secara *difusi*. Struktur *alveolus* terdiri dari selaput tipis dan terdapat banyak kapiler darah. Di *alveolus* darah akan melepaskan karbon dioksida ke udara dan mengambil oksigen dari udara.
4. *Pleura* adalah selaput yang melapisi paru-paru. Strukturnya seperti kantong serta halus dan licin. Fungsi *pleura* adalah untuk mengurangi gesekan saat paru-paru mengembang atau mengempis. *Pleura* terdiri dari dua lapisan yaitu *pleuraparietal* dan *pleuraviserat*. Terdapat sedikit cairan yang mengandung *glikosaminoglikan* diantara kedua lapisan tersebut.
5. *Diafragma* adalah otot berserat yang menjadi pembatas antara rongga dada dan rongga perut. *Diafragma* juga berperan penting dalam proses pernapasan perut. *Diafragma* tersusun atas otot, pembuluh darah, dan saraf yang disebut saraf *frenikus*.
6. *Trakea* adalah tabung dengan panjang sekitar 5 inci yang menghubungkan laring dengan *bronkus*. *Trakea* tersusun atas tulang rawan *hialin* berbentuk seperti huruf C dan dilapisi oleh *epitel* bersilia. Fungsi *trakea* adalah

sebagai saluran pernapasan. *Silia* yang terdapat dalam sel *epitel* berfungsi untuk menangkap partikel asing dan membawanya ke faring sehingga dapat masuk ke sistem pencernaan.

#### Fungsi Paru-Paru :

1. Paru-paru ialah organ *respirasi* untuk pertukaran gas pernapasan karbon dan oksigen *diaksida*.
2. Karena ketika bernapas karbon kita memancarkan limbah *dungsidiaksida* paru merupakan bagian dari sistem ekskresi.
3. Untuk mengontrol pH darah dengan mengubah suatu tekanan *karbondioksida*.
4. Untuk menyaring bekuan darah yang terbentuk di suatu *vena*.
5. Untuk mempengaruhi suatu konsentrasi beberapa zat biologis dan obat yang yang digunakan dalam pengobatan darah.
6. Untuk mengubah suatu *angiotensin I* menjadi *angiotensin II* oleh suatu enzim *angiotensin-converting*.
7. Untuk dapat berfungsi sebagai suatu lapisan pelindung *shock* jantung.
8. Untuk memberikan suatu aliran udara untuk membuat suara.
9. Fungsi paru-paru juga sebagai *reservoir* darah didalam tubuh. *Volume* darah paru ialah rata-rata sekitar 450 mililiter, dan sekitar 9 persen dari total *volume* darah dari seluruh sistem peredaran darah.
10. *Eskalator* ialah suatu sistem pertahanan yang penting terhadap infeksi udara ditanggung. Partikel debu dan bakteri di udara yang dihirup

terperangkap di suatu lapisan lendir hadir pada permukaan *mukosa* dari saluran pernapasan dan naik menuju faring oleh berirama ke atas pemukulan *silia*.

### **2.8.1 Tuberculosis**

*Tuberculosis* (TBC atau TB) adalah penyakit infeksi pada saluran pernafasan yang disebabkan oleh bakteri. Bakteri ini merupakan bakteri basil yang sangat kuat sehingga memerlukan waktu lama untuk mengobatinya. Bakteri ini lebih sering menginfeksi organ paru-paru (90%) dibandingkan bagian lain tubuh manusia, *Tuberculosis* (TBC) merupakan penyakit menular.

### **2.8.2 Gejala Penyakit TBC**

Penderita yang terserang basil tersebut biasanya akan mengalami demam tapi tidak terlalu tinggi yang berlangsung lama, biasanya dirasakan malam hari disertai keringat malam. Kadang-kadang serangan demam seperti *influenza* dan bersifat hilang timbul. Gejala lain, penurunan nafsu makan dan berat badan, batuk-batuk selama lebih dari 3 minggu (dapat disertai dengan darah), perasaan tidak enak (*malaise*), dan lemah.

#### ○ **Gejala Tuberculosis**

- Batuk terus menerus dan berdahak selama tiga pekan atau lebih.
- Dahak bercampur darah/batuk darah
- Sesak nafas dan rasa nyeri pada dada

- Demam/meriang lebih dari sebulan
- Berkeringat pada malam hari tanpa penyebab yang jelas
- Badan lemah dan lesu
- Nafsu makan menurun dan terjadi penurunan berat badan

Paling mudah untuk mengetahui seseorang terkena *tuberculosis* jika dia berkeringat pada malam hari tanpa penyebab yang jelas. Walaupun tidak bisa langsung ditetapkan tuberkulosis karena harus didiagnosis, tapi itu salah satu pertanda. Jika Anda lemas, batuk tak berhenti, nyeri pada dada, dan keringat pada malam hari, langsung segera periksa.

### **2.8.3 Penyebab infeksi TBC**

Penyakit ini diakibatkan infeksi kuman mikobakterium tuberkulosis yang dapat menyerang paru, ataupun organ-organ tubuh lainnya seperti kelenjar getah bening, usus, ginjal, kandung, tulang, sampai otak. TBC dapat mengakibatkan kematian dan merupakan salah satu penyakit infeksi yang menyebabkan kematian tertinggi.

TBC sangat mudah menular, yaitu lewat cairan di saluran napas yang keluar ke udara lewat batuk/bersin & dihirup oleh orang-orang di sekitarnya. Tidak semua orang yang menghirup udara yang mengandung kuman TBC akan sakit.

Pada orang-orang yang memiliki tubuh yang sehat karena daya tahan tubuh yang tinggi dan gizi yang baik, penyakit ini tidak akan muncul dan kuman TBC akan "tertidur". Namun, pada mereka yang mengalami kekurangan gizi, daya tahan

tubuh menurun/ buruk, atau terus-menerus menghirup udara yang mengandung kuman TBC akibat lingkungan yang buruk, akan lebih mudah terinfeksi TBC (menjadi 'TBC aktif') atau dapat juga mengakibatkan kuman TBC yang "tertidur" di dalam tubuh dapat aktif kembali (reaktivasi).

Infeksi TBC yang paling sering, yaitu pada paru, sering kali muncul tanpa gejala apa pun yang khas, misalnya hanya batuk-batuk ringan sehingga sering diabaikan dan tidak diobati. Padahal, penderita TBC paru dapat dengan mudah menularkan kuman TBC ke orang lain dan kuman TBC terus merusak jaringan paru sampai menimbulkan gejala-gejala yang khas saat penyakitnya telah cukup parah.

#### **2.8.4 Pengobatan penyakit TBC**

Untuk mendiagnosis TBC, dokter akan melakukan pemeriksaan fisik, terutama di daerah paru/dada, lalu dapat meminta pemeriksaan tambahan berupa foto *rontgen* dada, tes laboratorium untuk dahak dan darah, juga tes tuberkulin (*mantoux*/PPD). Pengobatan TBC adalah pengobatan jangka panjang, biasanya selama 6-9 bulan dengan paling sedikit 3 macam obat.

Kondisi ini diperlukan ketekunan dan kedisiplinan dari pasien untuk meminum obat dan kontrol ke dokter agar dapat sembuh total. Apalagi biasanya setelah 2-3 pekan meminum obat, gejala-gejala TBC akan hilang sehingga pasien menjadi malas meminum obat dan kontrol ke dokter.

Jika pengobatan TBC tidak tuntas, maka ini dapat menjadi berbahaya karena sering kali obat-obatan yang biasa digunakan untuk TBC tidak mempan pada kuman TBC (*resisten*). Akibatnya, harus diobati dengan obat-obat lain yang lebih mahal dan "keras". Hal ini harus dihindari dengan pengobatan TBC sampai tuntas.

Pengobatan jangka panjang untuk TBC dengan banyak obat tentunya akan menimbulkan dampak efek samping bagi pasien. Efek samping yang biasanya terjadi pada pengobatan TBC adalah nyeri perut, penglihatan/pendengaran terganggu, kencing seperti air kopi, demam tinggi, muntah, gatal-gatal dan kemerahan kulit, rasa panas di kaki/tangan, lemas, sampai mata/kulit kuning.

Itu sebabnya penting untuk selalu menyampaikan efek samping yang timbul pada dokter setiap kali kontrol sehingga dokter dapat menyesuaikan dosis, mengganti obat dengan yang lain, atau melakukan pemeriksaan laboratorium jika diperlukan.

Pengobatan untuk penyakit-penyakit lain selama pengobatan TBC pun sebaiknya harus diatur dokter untuk mencegah efek samping yang lebih serius/berbahaya.

Penyakit TBC dapat dicegah dengan cara:

- Mengurangi kontak dengan penderita penyakit TBC aktif.
- Menjaga standar hidup yang baik, dengan makanan bergizi, lingkungan yang sehat, dan berolahraga.

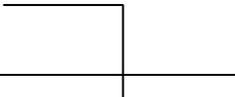
- Pemberian vaksin BCG (untuk mencegah kasus TBC yang lebih berat). Vaksin ini secara rutin diberikan pada semua balita.
- Perlu diingat bahwa mereka yang sudah pernah terkena TBC dan diobati, dapat kembali terkena penyakit yang sama jika tidak mencegahnya dan menjaga kesehatan tubuhnya.

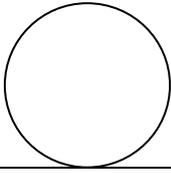
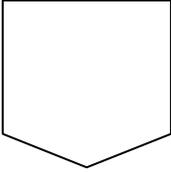
## 2.9 *Flowchart*

Bagan alir sistem (sistem *flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *Flowchart* adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. (Jogiyanto, 2009:263).

*Flowchart* disusun dengan simbol yang dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut:

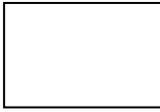
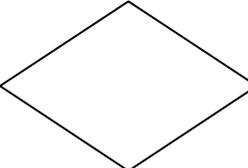
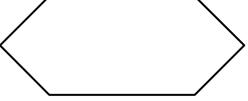
### 1. *FlowDirectionSymbols* (Simbol penghubung/alur)

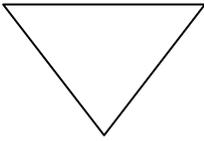
No	Simbol	Keterangan
1.		Simbol arus/ <i>flow</i> : untuk menyatakan jalannya arus suatu proses.
2.		Simbol <i>Communicationlink</i> : untuk menyatakan bahwa adanya transisi suatu

		data/informasi dari satulokasi ke lokasi lainnya.
3.		Simbol <i>Connector</i> : untuk menyatakan sambungandari satu proses ke proses lainnya dalamhalaman/lembar yang sama.
4.		Simbol <i>OfflineConnector</i> : untuk menyatakansambungan dari satu proses ke proses lainnya dalamhalaman/lembar yang berbeda.

Tabel 2.1 Tabel *FlowDirectionalSymbol*Sumber : Jogyanto (2009:796)

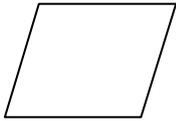
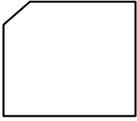
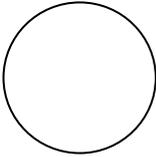
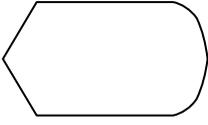
## 2. *ProcessingSymbols* (Simbol proses)

No	Simbol	Keterangan
1.		Simbol Proses: untuk menyatakan suatu tindakan(proses) tanpa tergantung pada jenis peralatannya.
2.		Simbol <i>Manual</i> : untuk menyatakan suatu tindakan(proses) yang tidak dilakukan oleh komputer.
3.		Simbol <i>Decision</i> : untuk menunjukkan suatu kondisitertentu yang akan menghasilkan dua kemungkinanjawaban, ya/tidak.
4.		Simbol <i>PredefinedProcess</i> : untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

5.		Simbol Terminal: untuk menyatakan permulaan atau akhir suatu program.
6.		Simbol <i>Keying Operation</i> : untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
7.		Simbol <i>off-line storage</i> : untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8.		Simbol <i>Manual input</i> : untuk memasukkan data secara <i>manual</i> dengan menggunakan online <i>keyboard</i> .

Tabel 2.2TabelProcessingSymbols.Sumber : Jogiyanto (2009:796)

### 3. *Input-Output Symbols* (Simbol masukan-keluaran)

No	Simbol	Keterangan
1.		Simbol <i>Input-output</i> : untuk menyatakan proses input dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
2.		Simbol <i>PunchedCard</i> : untuk menyatakan input berasal dari kartu atau <i>output</i> ditulis ke kartu.
3.		Simbol <i>Magnetic-tape</i> unit: untuk menyatakan input berasal dari pita magnetik atau <i>output</i> disimpan ke pita magnetik.
4.		Simbol <i>Disk storage</i> : untuk menyatakan input berasal dari disk atau <i>output</i> disimpan ke disk.
5.		Simbol <i>Document</i> : untuk mencetak laporan ke <i>printer</i> .
6.		Simbol <i>Display</i> : untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).

Tabel 2.3 Table *Processing Symbols*. Sumber : Jogiyanto (2009:796)

## 2.10 Data Flow Diagram

*Data Flow Diagram* (DFD) merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD menggambarkan arus data dalam sistem yang terstruktur dengan jelas.

Elemen dasar dari DFD yaitu:

1. Entitas Luar (*External Entity*)/Terminator

Entitas luar adalah entitas yang berada di luar sistem yang memberikan data kepada sistem atau yang menerima informasi dari sistem. Entitas ini disebut juga sumber atau tujuan data dan dianggap *eksternal* terhadap sistem yang sedang digambarkan. Bila suatu sistem informasi dirancang untuk satu bagian/ departemen maka bagian lain yang masih terkait menjadi entitas luar.



Gambar 2.6. Entitas Luar

Sumber: Ladjamudin (2009:265)

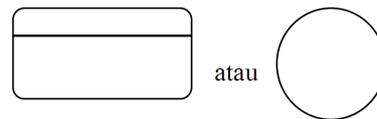
Pedoman pemberian nama entitas luar:

- 1) Nama entitas luar berupa kata benda
- 2) Entitas yang sama bisa digunakan lebih dari sekali atas suatu diagram aliran tertentu untuk menghindari persilangan antara jalur-jalur aliran data.

2. Proses (*Process*)

Proses menunjukkan suatu perubahan di dalam data. Jadi, aliran data yang meninggalkan suatu proses selalu diberi label yang berbeda dari aliran data yang masuk. Proses berfungsi mentransformasikan satu

atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa data masuk serta menghasilkan satu atau beberapa data keluaran.



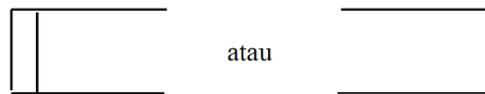
Gambar 2.7 Notasi Proses.

Sumber: Ladjamudin (2009:266)

Pedoman pemberian nama proses:

1. Nama proses berupa kata kerja atau kata benda secara detail yang mencerminkan fungsi proses tersebut, misalnya sistem kontrol inventaris, sistem hitung bonus.
  2. Jangan menggunakan kata proses sebagai bagian dari nama suatu proses.
  3. Tidak boleh ada beberapa proses yang memiliki nama yang sama.
  4. Proses harus diberi nomor. Urutan nomor sedapat mungkin mengikuti aliran atau urutan proses, namun demikian urutan nomor tidak secara mutlak merupakan urutan proses secara kronologis.
  5. Penomoran proses pada tingkat pertama (diagram nol) adalah 1.0, 2.0, 3.0, dan seterusnya.
  6. Diagram konteks tidak perlu diberi nomor.
3. Penyimpanan Data (*Data Store*)

Data Store merupakan tempat penyimpanan data yang ada dalam sistem. Data store dapat disimbolkan dengan bujur sangkar dengan salah satu ujung terbuka. Penyimpanan data menandakan penyimpanan manual seperti *file*/lemari, ditulis dengan sebuah kata benda.



Gambar 2.8 Notasi Data Store

Sumber: Ladjamudin (2009:267)

Pedoman pemberian nama data store:

- a. Nama harus mencerminkan data store tersebut.
- b. Bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

#### 4. Arus Data (*Data Flow*)

Arus data menunjukkan perpindahan data dari satu titik ke titik lain, dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses, data store dan menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil pemrosesan sistem.



Gambar 2.9 Notasi Arus Data Sumber: Ladjamudin (2009: 267)

Pedoman pemberian nama aliran data:

1. Nama aliran data yang terdiri dari beberapa aliran kata dihubungkan dengan garis sambung.
2. Tidak boleh ada aliran data yang namanya sama, dan pemberian nama harus mencerminkan isinya.
3. Aliran data yang terdiri dari beberapa elemen dapat dinyatakan dengan grup elemen.
4. Hindari penggunaan kata 'data' dan 'informasi' untuk memberi nama pada aliran data.
5. Sedapat mungkin nama aliran data ditulis lengkap.

Ketentuan lain:

1. Nama aliran data yang masuk ke dalam suatu proses tidak boleh samadengan nama aliran data yang keluar dari proses tersebut.
2. Aliran data yang masuk ke atau keluar dari data *store* tidak perlu diberi nama bila aliran data sederhana dan mudah dipahami serta aliran data menggambarkan seluruh data item (satu *record* utuh).
3. Tidak boleh ada aliran data dari entitas luar ke data store atau sebaliknya karena entitas luar bukan bagian dari sistem. Hubungan entitas luar dengan data store harus melalui proses. (Ladjamudin, 2009:67).

Ada 3 macam DFD, yaitu:

1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada data *store* dalam diagram konteks.

## 2. Diagram Nol/Zero (*Overview Diagram*)

Diagram nol adalah diagram yang menggambarkan lebih rinci proses utama yang ada pada diagram konteks, diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan entitas luar. Pada *level* ini sudah dimungkinkan adanya/digambarkannya data *store* yang digunakan. (Ladjamudin, 2009:64)

## 3. Diagram Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* dan diagram *level* di atasnya. (Ladjamudin, 2009:64).

## 2.11 ERD (*Entity Relationship Diagram*)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data. (Ladjamudin, 2009:64).

Diagram hubungan entitas atau yang lebih dikenal dengan sebutan ERdiagram, adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan dalam sistem secara abstrak.

Elemen-elemen ERD adalah sebagai berikut:

### 1. *Entity*

*Entity* adalah sebuah benda atau objek yang dapat dibedakan dari semua objek lainnya. *Entity* diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya). *Entity* digambarkan dengan sebuah bentuk persegi panjang.

### 2. *Atribut*

*Atribut* adalah properti atau karakteristik yang dimiliki oleh suatu entitas dimana properti atau karakteristik itu bermakna atau berarti bagi organisasi atau perusahaan. Setiap diagram hubungan entitas bisa terdapat lebih dari satu *atribut*. *Atribut* digambarkan dalam bentuk *elips*.

### 3. Hubungan Relasi (*Relationship*)

*Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*relationship*) diberi nama dengan kata

kerjadasar, sehingga memudahkan untuk melakukan pembacaan relasinya. Relasi digambarkan dalam bentuk belah ketupat.

#### 4. Kardinalitas

Kardinalitas relasi merujuk kepada hubungan maksimum yang terdiri dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat 3 macam kardinalitas relasi, yaitu :

##### 1. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang ke dua dan sebaliknya.

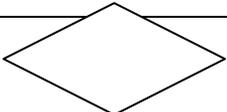
##### 2. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang ke dua.

##### 3. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.

Tabel Simbol yang digunakan dalam *Entity Relationship Diagram*

No	Simbol	Keterangan
1.		<i>Entity</i>
2.		<i>Relationship</i>

3.		<i>Attribut</i>
4.		<i>Link</i>

Tabel 2.4 Tabel *Entity Relationship Diagram*. (Ladjamudin, 2009:270).