BAB 2

TINJAUAN PUSTAKA

Bab ini berisi teori yang berhubungan dengan saham dan data mining baik penjelasan ataupun pemaparan metode yang dipilih. Teori mengenai saham dijelaskan di awal yang meliputi penjelasan mengenai definisi saham dan harga saham serta faktor-faktornya. Selanjutnya akan diuraikan tentang konsep Data mining yaitu defnisi dan klasifikasinya serta dijelaskan tentang penggunaan metode dan teknik tang dipakai.

2.1 Saham

Fund Manager adalah seorang profesional terlatih yang memiliki tanggung jawab dalam mengambil keputusan yang terkait dengan portofolio investasi[5]. Investasi yan dilakukan dapat berupa *mutual fund, pension fund atau insurance fund, Mutual Fund* merupakan sebuah bentuk pencarian dana yang dilakukan sebuah perusahaan dengan cara menjual kepemilikannya kepada public seperti *bond fund, equity fund, sector fund, spesiality fund, tax-fee bond fund,* dan *stock fund.*

Melalui berbagai jenis investasi yang ada, seorang *fund manager* harus memaksimalkan keuntungan yang diperoleh dari investasi tersebut. Salah satu jenis investasi yang paling marak dilakukan adalah *stock fund*. *Stock fund* merupakan jenis *mutual fund* yang memfokuskan investasinya dalam bentuk jual – beli saham. Saham adalah Surat berharga berupa bukti kepemilikan atas perusahaan. Dengan memiliki saham berarti seseorang akan mempunyai hak

dalam pemilihan dewan direksi perusahaan dan pembagian keuntungan. Investasi *stock fund* dapat menghasilkan keuntungan berupa[13] :

a. Pembagian *Dividen*, yaitu bagian dari keuntungan bersih perusahaan yang dibagikan kepada pemilik saham
b. *Capital gain*, yaitu selisih harga jual dan harga beli. Disebut juga selisih kenaikan kurs, jika pemilik saham menjual sahamnya dengan kurs yang lebih tingi dibandingkan dengan kurs pada waktu membelinya. Misalnya sewaktu membeli nilainya Rp 2000/saham dan kemudian dijual dengan harga Rp 2500/saham. Selisih sebesar Rp 500 disebut *capital gain.* Jika selisih harga jual dan harga beli negatif, maka yang terjadi adalah *Capital loss.*

Investasi jenis ini banyak dipilih karena dibandingkan investasi lainnya, saham memungkinkan pemiliknya mendapatkan keuntungan (*return*) yang lebih besar dalam waktu relatif singkat (*high return)*. Namun di samping *high* return, saham juga memiliki sifat *high risk* karena suatu ketika harga saham bisa menurun dengan cepat. Untuk memperoleh keuntungan yang besar, maka seorang *fund manager* harus dapat mengambil keputusan yang tepat untuk menjual atau membeli saham suatu perusahaan.

Saham terdiri dari dua macam yaitu :

1. *Common stock*

   Common Stock merupakan jenis saham yang umum diperjualbelikan di bursa saham. Saham jenis ini merupakan saham yang paling banyak dipegang oleh publik. Pemilik saham ini

memiliki hak suara dalam pengambilan keputusan perusahaan dan dalam pemilihan pimpinan perusahaan. Harga saham jenis ini cenderung berfluktuasi sehingga lebih sulit dianalisis dan dinilai. Pemilik *common stock* memperoleh keuntungan dari perusahaan yang mengeluarkan saham dalam bentuk saham dalam bentuk *capital gain* dan *dividen[11]*. Harga *common stock* dapat meningkat tajam sehingga memberikan *capital gain* yang besar untuk pemegang saham. Biasanya harga saham dapat naik drastis jika performa perusahaan baik. Dengan *capital gain* yang besar maka pemilik *common stock* berpotensi mendapat keuntungan lebih tinggi. Namun *common stock* juga berpotensi mendatangkan resiko yang tinggi jika yang terjadi adalah *capital loss*.

Dividen yang dibayarkan kepada pemilik saham diambil dari keuntungan bersih perusahaan. Pembayaran *dividen* kepada pemilik *common stock* bergantung kepada pemilik *common stock* bergantung pada kondisi perusahaan sehingga tidak dijamin bahwa *dividen* akan diberikan secara rutin. Pembagian *dividen* bagi pemilik *common stock* dilakukan setelah pihak – pihak lain menerima bagian masing-masing. Demikian pula halnya jika perusahaan dilikuidasi, pemilik *common stock* berhak menerima aset perusahaan setelah perusahaan memenuhi pembagian aset pihak lain. Pihak lain tersebut dalam hal ini antara lain pemegang *bond stock* dan pemilik *preferred stock.* Oleh karena itu, kepemilikan *common stock* bersifat *residual*, karena

pemilik *common stock* berhak atas pendapatan maipin aset perusahaan setelah bagian untuk pihak lain diberikan.

2. *Preferred stock*

Seperti *common stock*, *preferred stock* dijual oleh perusahaan dan diperdagangkan bagi para investor di bursa saham. Karakteristik utama dari *preferred stock* adalah harganya cenderung stabil. Karena harganya yang stabil, maka *capital gain* yang diharapkan tidak besar sehingga tidak dapat dijadikan sumber keuntungan utama. Harga yang stabil juga menyebabkan *capital loss* yang kecil sehingga resiko yang ditimbulkan juga kecil. Karakteristik lain dari *preferred stock* adalah *dividen* yang tetap dan rutin. Pembayaran *dividen* kepada pemilik *preferred stock* dilakukan secara rutin selama kepemilikan saham tersebut. Pemilik *preferred stock* tidak memperoleh keuntungan sama dengan pemilik *common stock[3]*. Jika performa perusahaan baik, harga *common stock* dapat meningkat dan memberikan *capital gain* dan *dividen* yang besar. Namun tidak demikian halnya dengan pemilik *preferred stock* yang menerima *dividen* dengan jumlah tetap walaupun performa baik. Pembagian *dividen* bagi pemilik *preferred stock* dilakukan setelah pemegang *bond* menerima bagiannya. Demikian pula jika perusahaan dilikuidasi maka pemilik *preferred stock* akan menerima pembagian kekayaan setelah pemegang *bond* dan lebih dulu dari pemilik *common stock*.

Pemegang *preferred stock* cenderung melakukan investasi yang stabil melalui pembayaran *dividen* yang tepat dan tidak bergantung pada kondisi pasar dan perusahaan sehingga investor cenderung untuk mencari perusahaan yang memiliki keuntungan besar dan stabil.

Pemilihan jenis saham yang akan dibeli bergantung pada kondisi yang diinginkan oleh investor. Jika investor menginginkan investasi yang menghasilkan keuntungan yang besar dan cepat maka investor cenderung membeli *common stock,* namun jika investor menginginkan investasi yang beresiko rendah dengan keuntungan stabil maka investor cenderung membeli *preferred stock.*

2.1.1 Harga Saham

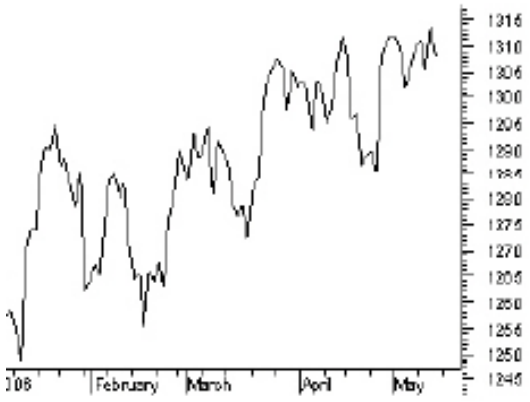
Saham merupakan surat berharga yang memberikan hasil tidak tetap sehingga penetapan harganya cukup sulit [13]. Harga saham akan selalu mengalami perubahan baik kenaikan maupun penurunan. Untuk itu diperlukan pendekatan untuk menganalisa harga saham. Ada dua macam pendekatan untuk menganalisa harga saham, yaitu

1. Analisis Teknikal

   Analisis teknikal merupakan suatu metodologi untuk memprediksi fluktuasi harga saham berdasarkan kondisi penawaran dan permintaan akan saham tersebut. Kondisi ini ditentukan oleh berbagai faktor mulai dari faktor ilmiah sampai opini dan pendapat. Fluktuasi harga saham ini akan membentuk tren. Asumsi yang diambil dalam analisis

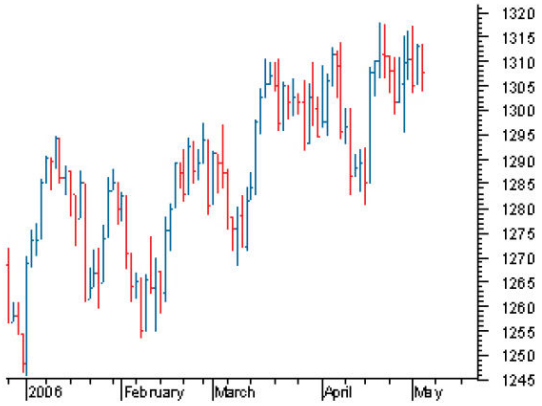
teknikal antara lain bahwa nilai saham semata-mata dipengaruhi oleh penawaran dan permintaan[13], harga saham cenderung bergerak pada suatu tren yang tetap untuk jangka waktu yang lama dan perubahan terhadap tren terjadi karena pergeseran penawaran dan permintaan. Analis teknikal mempelajari perubahan harga saham dengan menggunakan tiga dasar diagram yaitu[13] :

a. *Line Chart* (diagram garis), diagram ini hanya mengikuti harga saham dan hanya merepresentasikan harga saham penutupan.



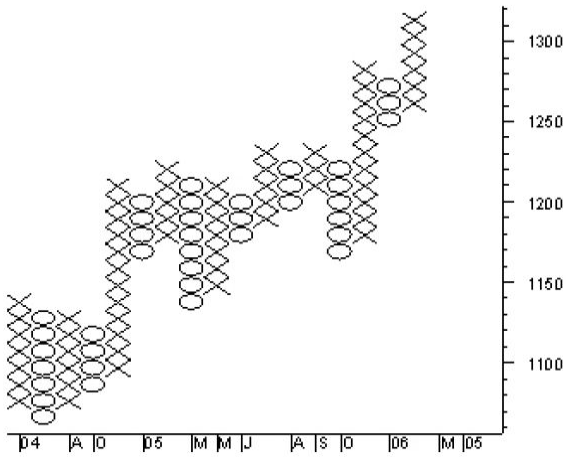
Gambar 2.1 Diagram garis dalam Analisis Teknikal

b. *Bar Charts (*diagram batang), terdiri dari garis yang tidak putus-putus yang menggambarkan suatu harga. Suatu garis vertical (|) menggambarkan tinggi rendahnya harga yang diperdagangkan setiap harinya. Titik pada setiap batang (-) menggambarkan harga penutupan pada hari tersebut.



Gambar 2.2 Diagram Batang dalam Analisi Teknikal

c. *Point and Figure charts* (diagram titik dan bentuk). Diagram ini tidak hanya digunakan untuk melacak pengulangan – pengulangan dari suatu tren, melainkan juga dapat digunakan untuk perkiraan harga yang terjadi di kotak target harga. X merepresentasikan tren kenaikan harga dan O merepresentasikan tren penurunan harga.



Gambar 2.3 Diagram Titik dan Bentuk dalam Analisis Teknikal

2. Analisis Fundamental

Ide Dasar dari Analisis fundamental ini adalah bahwa saham memiliki nilai intrinsik yaitu harga saham yang sesungguhnya sebagaimana diperkirakan oleh investor. Analisi fundamental ini mempertimbangkan berbagai faktor yang dianalisi yaitu :

a. Analisis makroekonomi

Analisis makroekonomi merupakan hal yang sangat penting dilakukan oleh investor karena kondisi makroekonomi mempengaruhi operasi perusahaan sehari-hari. Analisis terhadap indikator makroekonomi memingkinkan investor memahami kondisi makroekonomi di masa datang yang sangat berguna dalam pembuatan keputusan investasi yang menguntungkan. Indikator makroekonomi yang perlu diamati adalah pertumbuhan ekonomi, nilai tukar rupiah, laju inflasi, tingkat bunga, dan cadangan devisa[3]. Kondisi makroekonomi menggambarkan iklim investasi sehingga mempengaruhi minat investor untuk berinvestasi. Adapun kondisi makroekonomi yang menggambarkan iklim investasi yang baik ditunjukan oleh nilai laju inflasi yang rendah, suku bunga yang rendah, niali tukar rupiah terhadap US Dollar yang kuat, cadangan devisa Negara yang besar dan pertumbuhan ekonomi yang tinggi. Stabilisasi makroekonomi, khususnya inflasi rendah, penyaluran kredit berkelanjutan dan nilai tukar yang realistik merupakan dasar bagi berfungsinya pasar keuangan yang efektif.

Laju inflasi menunjukan daya beli atau nilai uang [4]. Inflasi sendiri adalah kecenderungan terjadinya oeningkatan harga – harga produk secara keseluruhan. Jika laju inflasi tinggi maka harga produk meningkat sehingga menyebabkan daya beli uang menurun. Sementara jika laju inflasi rendah, daya beli uang terhadap produk meningkat sehingga pendapatan perusahaan bertambah dan dapat meningkat pendapatan riil yang diperoleh investor dari investasinya jika membeli saham perusahaan tersebut.

Laju inflasi berhubungan dengan suku bunga. Laju inflasi yang tinggi menyebabkan meningkatkan jumlah uang yang ada di masyarakat sehingga untuk menarik uang di masyarakat, suku bunga ditingkatkan. Suku bungan yang tingi menyebabkan kesempatan investasi yang ada tidak menarik lagi karena meningkatkan biaya modal yang harus ditanggung perusahaan.

Nilai tukar rupiah terhadap US Dollar yang menguat merupakan sinyal positif bagi perekonomian [3]. Nilai tukar mempunyai peran penting dalam rangka tercapainya stabilitas moneter dan dalam mendukung kegiatan ekonomi. Nilai tukar yang stabil diperlukan untuk terciptanya iklim yang kondusif bagi peningkatan kegiatan dunia usaha.

Pertumbuhan ekonomi menunjukan daya beli masyarakat. Jika pertumbuhan ekonomi membaik maka daya beli masyarakat meningkat. Hal ini merupakan kesemparan bagi perusahaan untuk

meningkatkan penjualan yang pada akhirnya dapar meningkatkan keuntungan.

b. Analisis industri

Industri dimana perusahaan berada mempengaruhi masa depan perusahaan tersebut. Biasanya saham yang lemah dalam industri yang kuat lebih disukai daripada saham yang kuat dalam industri yang lemah. Dalam analisis industri, investor cenderung mencoba membandingkan kinerja dari berbagai industri untuk bisa mengetahui jenis industri apa saja yang memberikan prospek paling menjanjikan ataupun sebaliknya. Keseluruhan faktor ini dapat mempengaruhi harga saham suatu perusahaan.

c. Analisis perusahaan

Fokus perhatian terbesar dalam analisis fundamental adalah analisis terhadap perusahaan itu sendiri. Analisis ini bertujuan untuk menentukan harga pasar yang wajar (nilai instrisik) terhadap saham. Harga pasar saham dibandingkan dengan nilai intrinsiknya, jika harga pasarnya lebih rendah dari nilai intrinsiknya (*undervalued)* maka saham tersebut layak dibeli, tetapi sebaliknya jika harga pasar lebih tinggi dari nilai intrinsiknya (*overvalued),* saham tersebut layak dijual. Hal yang harus diperhatikan oleh investor dalam analisis perusahaan adalah pendapatan perusahaan dan *dividen* yang akan diterima investor. Karena *dividen* diambil

dari pendapatan perusahaan. Sehingga pendapatan perusahaan adalah kunci dari analisis fundamental.

2.1.2 Penentuan Harga Saham Intrisik dengan Indikator EMA

Eksponensial Moving Average (EMA): Memberikan nilai pembanding (weightage) lebih besar pada nilai terbaru dengan tidak membuang pengamatan historical data secara keseluruhan data dengan analisa menggunakan teknikal EMA. Dalam sebuah penelitian, EMA dianggap sebagai indikator utama karena kemampuannya untuk menangani jumlah hampir tak terbatas dari data masa lalu, suatu sifat yang sangat berharga dalam deret waktu prediksi (Perlu dicatat bahwa penerapan indikator lain mungkin menghasilkan yang lebih baik akurasi prediksi untuk saham yang dipertimbangkan)[5].

EMA (t) = EMA (t-1) + alpha * (Harga (t) - EMA (t-1))

Dimana, alpha = 2 / (N +1), Jadi, untuk N = 9, alpha = 0,20 Dalam teori, Bursa Prediksi Soal dapat dianggap sebagai mengevaluasi fungsi F pada T waktu berdasarkan nilai-nilai sebelumnya dari F pada waktu t-1, t-2, tn sementara menempatkan berat sesuai fungsi w di setiap titik untuk F. F (t) = w1 * F (t-1) + w2 * F (t-2) + ... + w * F (t-n).

Berikut ini adalah contoh dari pembentukan titik nilai dari menggunakan indicator teknis EMA terhadap 50 sample pada Emiten ISAT. Dengan diasumsikan pola dibentuk dengan parameter ‘Close’.

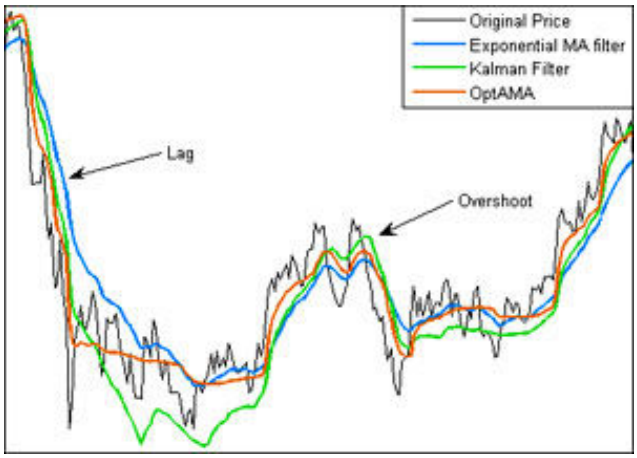
Untuk memilih parameter sebagai perhitungan pembentukan Pola EMA (Exponential Moving Average), nilai EMA bisa dihitung menggunakan rumus berikut

$$XMA = (2/(periode + 1)\, x(Current\ Price - PreviousXMA)) + PreviousXMA$$

Dengan

$$PreviousXMA = \frac{Price_n + Price_{n-1}}{n}$$

Dari Sample data tersebut akan menghasilkan sebuah pola dan Pola XMA yang menjadi acuan fluktuasi Harga Saham terbentuk, kecenderungan inilah yang menjadi *decission* point dengan melihat grafik perubahan, kecenderungan XMA nilai akhir yang menjadi tolok ukur apakah cenderung naik, atau turun.



Gambar 2.4 Sample Grafik Pembentukan pembentukan titik nilai EMA

2.2 Data Mining

2.2.1 Pengertian *Data Mining*

Data Mining (DM) adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari *database* skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Definisi umum dari DM itu sendiri adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data[8].

Data mining adalah kombinasi secara logis antara pengetahuan data, dan analisa statistik yang dikembangkan dalam pengetahuan bisnis atau suatu proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, tiruan dan *machine-learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat bagi pengetahuan yang terkait dari berbagai database besar [14].

Data mining meliputi tugas-tugas yang dikenal sebagai ekstraksi pengetahuan, arkeologi data, eksplorasi dalam pemrosesan pola data dan memanen informasi. Semua aktifitas ini dilakukan secara otomatis dan mengizinkan adanya penemuan cepat bahkan oleh non programmer[14].

Data mining cerdas menemukan informasi di dalam data *warehouse* dimana laporan dan query tidak bisa diungkapkan secara efektif. Piranti *data mining* menemukan pola-pola di dalam data dan bahkan menyimpulkan aturan dari data tersebut (Therling K., 2006)[14].

Data mining (DM)[6] yang juga dikenal sebagai *Knowledge Discovery* (Frawley et al., 1992), merupakan salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari *database* skala besar yang makin banyak terakumulasi sejalan dengan pertumbuhan teknologi informasi. Secara umum, *data mining* dapat didefinisikan sebagai suatu rangkaian proses untuk menggali nilai tambah berupa ilmu pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data [5].

Definisi umum dari *Data Mining* adalah proses pencarian pola-pola yang menarik (*hidden pattern*) berupa pengetahuan (*knowledge*) yang tidak diketahui sebelumnya dari suatu kumpulan data dimana data tersebut dapat berada dalam *database, data warehouse*, atau media penyimpanan informasi yang lain. *Data Mining* merupakan proses analisis terhadap data dengan penekanan menemukan informasi yang tersembunyi pada sejumlah besar data yang disimpan ketika menjalankan bisnis perusahaan.

2.2.2 Tujuan *Data mining*

Dengan kemampuan *Data mining* (penambangan data) untuk mencari informasi bisnis yang berharga dari basis data yang sangat besar, yang dilakukan dapat dianalogikan dengan penambangan logam mulia dari lahan sumbernya, teknologi ini dipakai untuk :

1. Prediksi trend dan sifat-sifat bisnis.

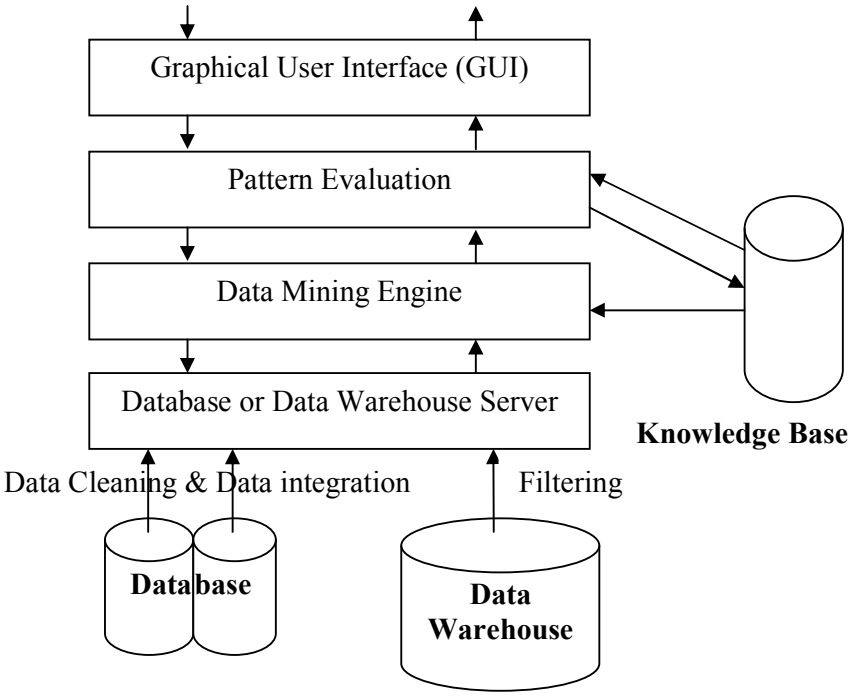
   Data mining mengotomatisasi proses pencarian informasi pemprediksi di dalam basis data yang besar. Pertanyaan-pertanyaan yang berkaitan dengan prediksi ini dapat cepat dijawab langsung dari data yang tersedia. Contoh dari masalah prediksi ini misalnya target pemasaran, peramalan kebangkrutan, dan bentuk-bentuk kerugian lainnya.

2. Penemuan pola-pola yang tidak diketahui sebelumnya.

   Kakas *data mining* "menyapu" basis data, kemudian mengidentifikasi pola-pola yang sebelumnya tersembunyi dalam satu sapuan. Contoh dari penemuan pola ini adalah analisis pada data penjulan ritel untuk mengidentifikasi produk-produk, yang kelihatannya tidak berkaitan, yang seringkali dibeli secara bersamaan oleh customer.

2.2.3 Tahapan *Data Mining*

Dalam aplikasinya, *data mining* sebenarnya merupakan bagian dari proses *Knowledge Discovery in Database* (KDD), bukan sebagai teknologi yang utuh dan berdiri sendiri. Data mining merupakan suatu bagian langkah yang penting dalam proses KDD terutama berkaitan dengan ekstraksi dan penghitungan pola-pola dari data yang ditelaah, seperti ditunjukan oleh gambar 2.5 dibawah ini :[8]



Gambar 2.5 Tahapan pada proses *Data Mining*

a. Data cleaning

Untuk menghilangkan data *noise* (data yang tidak relevan/berhubungan langsung dengan tujuan akhir proses *data mining*, misal: *data mining* yang bertujuan untuk menganalisa hasil penjualan, maka data-data dalam kumpulan seperti ”nama pegawai”, ”umur”, dan sebagainya dapat di-*ignore*) dan tidak konsisten.

b. Data integration

Untuk menggabungkan *multiple data source.*

c. *Data selection*

Untuk mengambil data yang sesuai untuk keperluan analisa.

d. *Data transformation*

Untuk mentransformasikan data ke dalam bentuk yang lebih sesuai untuk di *mining.*

e. *Data mining*

Proses terpenting dimana metode tertentu diterapkan untuk menghasilkan *data pattern*.

f. *Pattern evaluation*

Untuk mengidentifikasi apakah *interenting patterns* yang didapatkan sudah cukup mewakili *knowledge* berdasarkan perhitungan tertentu.

g. *Knowledge presentation*

Untuk mempresentasikan *knowledge* yang sudah didapatkan dari *user.*

2.2.4 Teknik Klasifikasi (Classification)

Klasifikasi adalah suatu fungsionalitas *data mining* yang akan menghasilkan model untuk memprediksi kelas atau kategori dari objek-objek di dalam basis data. Klasifikasi merupakan proses yang terdiri dari dua tahap, yaitu tahap pembelajaran dan tahap pengklasifikasian.

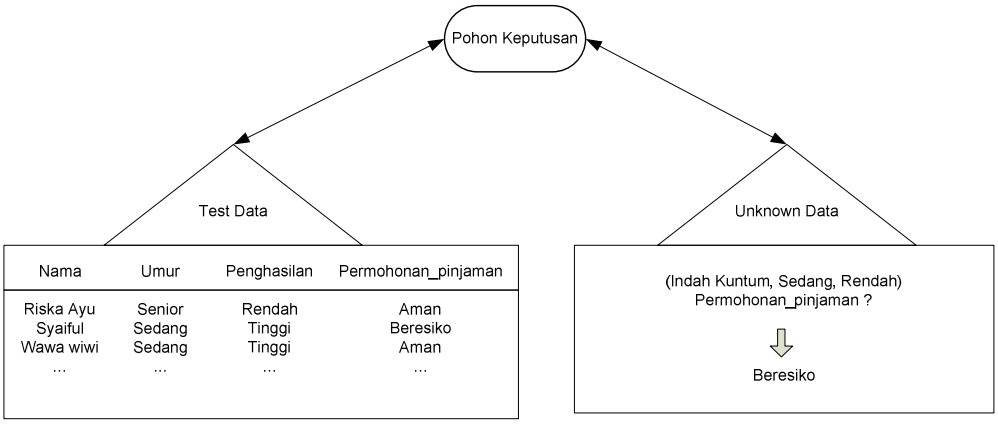
Pada tahap pembelajaran, sebuah algoritma klasifikasi akan membangun sebuah model klasifikasi dengan cara menganalisis training data. Tahap pembelajaran dapat juga dipandang sebagai tahap pembentukan fungsi

atau pemetaan Y=F(X) di mana Y adalah kelas hasil prediksi dan X adalah tuple yang ingin diprediksi kelasnya.

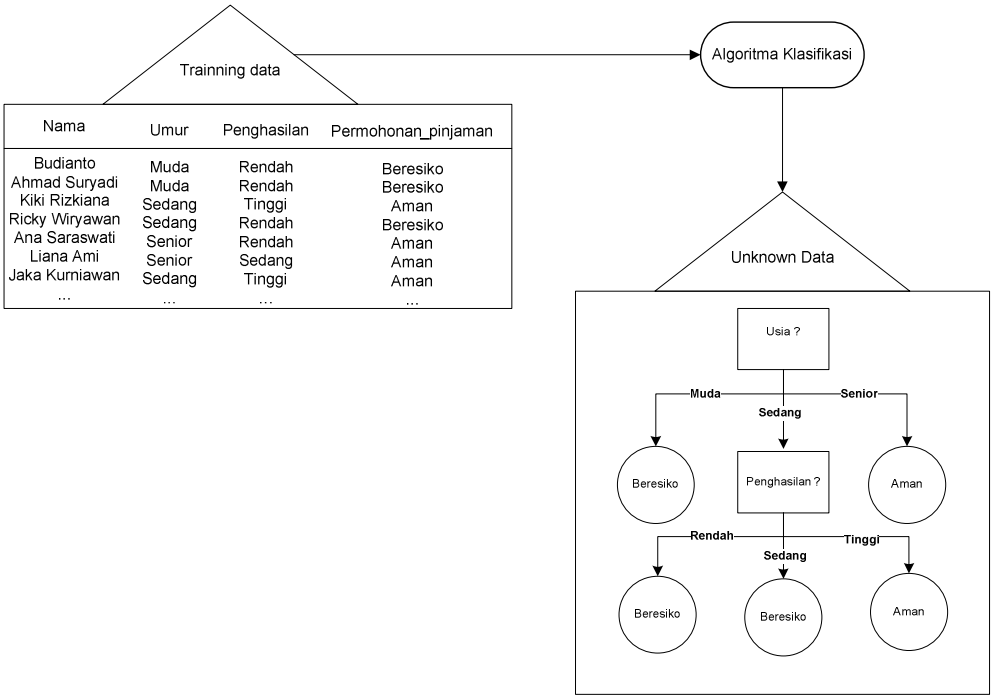
Selanjutnya, pada tahap pengklasifikasian, model yang telah dihasilkan akan digunakan untuk melakukan klasifikasi terhadap *unknown data*. Akan tetapi, sebuah model hanya boleh digunakan untuk klasifikasi jika akurasi model tersebut cukup tinggi. Akurasi dapat diketahui dengan cara menguji model tersebut dengan *test data*. *Test data* terdiri dari *tuple-tuple* yang kelasnya sudah diketahui, namun *test data* tidak boleh sama dengan *training data* karena akan menyebabkan pengujian tersebut menunjukkan akurasi yang tinggi, padahal belum tentu .

2.2.5 Pohon Keputusan (*decision tree*)

Pohon keputusan[10] merupakan salah satu bentuk penggambaran model klasifikasi. Pada pohon keputusan, simpul dalam menyatakan pengujian terhadap suatu atribut (digambarkan dengan kotak), cabang menyatakan hasil dari suatu pengujian (digambarkan dengan panah yang memiliki label dan arah), sementara daun menyatakan kelas yang diprediksi (digambarkan dengan lingkaran). Contoh pohon keputusan untuk kasus permohonan pinjaman dapat dilihat pada gambar 2.6. Pada saat membangun pohon keputusan, digunakan *attribute selection measures* dalam memilih kriteria terbaik untuk mempartisi *tuple-tuple* data ke dalam kelas-kelas berbeda. Kriteria tersebut meliputi *splitting attribute, split point,* maupun *splitting subset* .



Gambar 2.6 Tahap Pengklasifikasian



Gambar 2.7 Tahap Pembelajaran

2.2.6 Algoritma C4.5

Salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan[10]. Dalam prosedur algoritma ID3, input berupa sampel training, label training dan

atribut. Algoritma C4.5 merupakan pengembangan dari ID3. Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut [10]:

1. Pilih atribut sebagai *root*
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai gain tertinggi dari atribut-atribut yang ada. Untuk menghitung gain digunakan rumus seperti tertera dalam rumus 1

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy\ (S_i) \ldots\ldots\ldots\ldots\ (1)$$

Dengan,

S : Himpunan kasus
A : Atribut
n : Jumlah partisi A
|Si| : Jumlah kasus pada partisi ke I
|S| : Jumlah kasus dalam S

Sedangkan untuk perhitungan nilai entropy dapat dilihat pada rumus 2

$$Entropy(S) = \sum_{i=1}^{n} - pi * log2pi \ldots\ldots\ldots\ldots\ldots\ldots\ (2)$$

dengan,

S : Himpunan kasus
A : Fitur
n : Jumlah partisi S
pi : Proporsi dari Si terhadap S

2.2.7 Contoh Kasus

Untuk memudahkan penjelasan mengenai algoritma C4.5 berikut ini disertakan contoh kasus yang dituangkan dalam Tabel 2.1 [10]

Tabel 2.1 Tabel Keputusan Bermain Tenis

No	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Dalam kasus yang tertera pada Tabel 2.1, akan dibuat pohon keputusan untuk menentukan main tenis atau tidak dengan melihat keadaan cuaca, temperatur, kelembaban dan keadaan angin. Adapun langkah-langkahnya sebagai berikut :

1. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *OUTLOOK*, *TEMPERATURE*, *HUMIDITY* dan *WINDY*. Setelah itu lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.2

Tabel 2.2 Perhitungan Node 1

NODE	Parameter	Nama Parameter	Jumlah kasus (s)	Tidak (S_1)	Tidak (S_2)	Entropy	Gain
1	Total		14	4	10	0.86312069	
	Outlook						0.258521037
		Cloudy	4	0	4		
		Rainy	5	1	4	0.721928095	
		Sunny	5	3	2	0.970950594	
	Temperatur						0.18380925
		Cool	4	0	4	0	
		Hot	4	2	2	1	
		Mild	6	2	4	0.91829834	
	Humidity						0.37006501
		High	7	4	3	0.98228136	
		Normal	7	0	7	0	
	Windy						0.005977711
		False	8	2	6	0.811278124	
		True	6	4	2	0.91829834	

Baris TOTAL kolom *Entropy* pada Tabel 2.2 dihitung dengan rumus 2, sebagai berikut:

$$\text{Entropy(Total)} = \left(-\frac{4}{14} * \log 2\left(\frac{4}{14}\right)\right) + \left(-\frac{10}{14} * \log 2\left(\frac{10}{14}\right)\right)$$

$$\text{Entropy(Total)} = 0.863120569$$

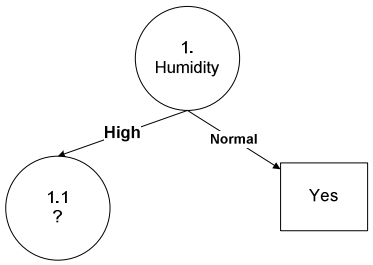
Sementara itu nilai Gain bari Outlook dihitung dengan menggunakan rumus 1, sebagai berikut:

$$\text{Gain(total, Outlook)} = Entropy(Total) - \sum_{i=1}^{n} \left|\frac{Outlook_i}{Total}\right| * Entropy(Outlook_i)$$

$$\text{Gain(total, Outlook)} = 0.863120569 - \left(\left(\frac{4}{14} * 0\right) + \left(\frac{5}{14} * 0.723\right) + \left(\frac{5}{14} * 0.97\right)\right)$$

$$\text{Gain(total, Outlook)} = 0.23$$

Dari hasil pada Tabel 2.2 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *HUMIDITY* yaitu sebesar 0.37. Dengan demikian *HUMIDITY* dapat menjadi node akar. Ada 2 nilai atribut dari *HUMIDITY* yaitu *HIGH* dan NORMAL. Dari kedua nilai atribut tersebut, nilai atribut NORMAL sudah mengklasifikasikan kasus menjadi 1 yaitu keputusannya *Yes*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *HIGH* masih perlu dilakukan perhitungan lagi. Dari hasil tersebut dapat digambarkan pohon keputusan sementaranya seperti tampak pada Gambar 2.8 berikut,



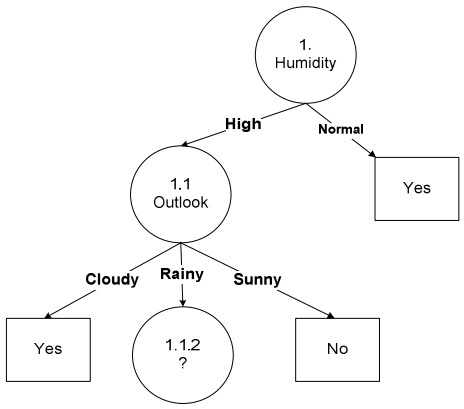
Gambar 2.8 Pohon Keputusan Hasil Perhitungan Node 1

2. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *OUTLOOK*, *TEMPERATURE* dan *WINDY* yang dapat menjadi node akar dari nilai atribut *HIGH*. Setelah itu lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungannya ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Perhitungan Node 1.1

NODE	Parameter	Nama Parameter	Jumlah kasus (s)	Tidak (S_1)	Tidak (S_2)	Entropy	Gain
1	Humidity-High		7	4	3	0.985228136	
	Outlook						0.69951385
		Cloudy	2	0	4	0	
		Rainy	2	1	4	1	
		Sunny	3	3	2	0	
	Temperatur						0.020244207
		Cool	0	0	4	0	
		Hot	3	2	2	0.91829834	
		Mild	4	2	4	1	
	Windy						0.020244207
		False	4	2	2	1	
		True	3	2	1	0.91829834	

Dari hasil pada Tabel 2.3 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *OUTLOOK* yaitu sebesar 0.67. Dengan demikian *OUTLOOK* dapat menjadi node cabang dari nilai atribut *HIGH*. Ada 3 nilai atribut dari *OUTLOOK* yaitu *CLOUDY*, *RAINY* dan *SUNNY*. Dari ketiga nilai atribut tersebut, nilai atribut *CLOUDY* sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya *Yes* dan nilai atribut *SUNNY* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *RAINY* masih perlu dilakukan perhitungan lagi. Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 2.9 berikut,



Gambar 2.9 Pohon Keputusan Hasil Perhitungan Node 1.1

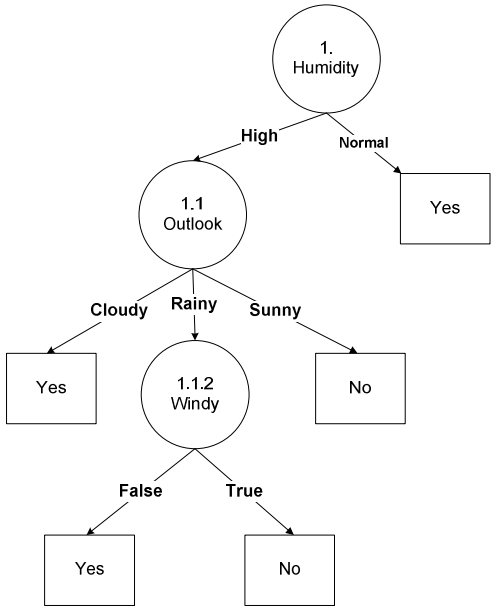
3. Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *TEMPERATURE* dan *WINDY* yang dapat menjadi node cabang dari nilai atribut *RAINY*. Setelah itu melakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.4.

Tabel 2.4 Perhitungan Node 1.1.2

NODE	Parameter	Nama Parameter	Jumlah kasus (s)	Tidak (S_1)	Tidak (S_2)	Entropy	Gain
1.1.2	Humidity-High and Outlook Rainy		2	1	1	1	
	Outlook						0
		Cool	0	0	0	0	
		Hot	0	0	0	0	
		Mild	2	1	1	1	
	Windy						1
		False	1	0	1	0	
		True	1	1	1	1	

Dari hasil pada tabel 2.4 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *WINDY* yaitu sebesar 1. Dengan demikian *WINDY*

dapat menjadi node cabang dari nilai atribut *RAINY*. Ada 2 nilai atribut dari *WINDY* yaitu *FALSE* dan *TRUE*. Dari kedua nilai atribut tersebut, nilai atribut *FALSE* sudah mengklasifikasikan kasus menjadi 1 yaitu keputusan-nya *Yes* dan nilai atribut *TRUE* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut untuk nilai atribut ini. Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 2.10.



Gambar 2.10 Pohon Keputusan Hasil Perhitungan Node 1.1.2

Dengan memperhatikan pohon keputusan pada Gambar 2.6, diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 2.6 merupakan pohon keputusan terakhir yang terbentuk.

2.3 Konsep Dasar Data dan Informasi

2.3.1 Pengertian Data

Menurut Azhar Susanto : “Data adalah fakta atau apapun yang dapat digunakan input dan menghasilkan informasi”. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan kenyataan. Data merupakan suatu istilah yang berbentuk jamak dari kata “*datum*” yang berarti fakta atau bagian dari fakta yang mengandung arti yang menghubungkan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata, angka-angka, huruf-huruf yang menunjukan suatu ide, objek, kondisi dan situasi.

Menurut *the liang gie*: ”Data atau bahan keterangan adalah hal atau peristiwa kenyataan lainnya apapun yang mendukung suatu pengetahuan untuk dijadikan dasar guna penyusunan keterangan pembuatan kesimpulan atau penetapan keputusan, atau data ibarat bahan mentah yang melalui pengolahannya tertentu lalu menjadi keterangan (informasi)”.

2.3.2 Konsep Dasar Informasi

Informasi adalah hasil pengolahan data yang memberikan arti dan manfaat. Sumber dari informasi adalah data. Data bisa disebut juga kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Seperti halnya informasi didalam sebuah perusahaan yang sangat penting untuk mendukung kelangsungan perkembangannya, sehingga terdapat alasan bahwa informasi sangat

dibutuhkan bagi sebuah perusahaan. Akibatnya bila kurang mendapat informasi, dalam waktu tertentu perusahaan akan mengalami ketidak mampuan mengontrol sumber daya, sehingga dalam mengambil keputusan-keputusan strategis sangat terganggu, yang pada akhirnya akan mengalami kekalahan dalam bersaing dengan lingkungan pesaingnya.

2.3.2.1 Nilai Informasi

Nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan di dalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak di dalam perusahaan.

Hal-hal yang mendukung suatu informasi yaitu :

a. Data

   Data adalah inti dari sistem informasi. Setiap informasi mengumpulkan, memproses dan memelihara perbedaan yang luas dari data berkisar pada kesatuan ketertarikan terhadap hal itu. Komponen data dikelompokkan ke dalam tiga tipe : *input* data dimasukkan ke dalam sistem, penyimpanan data dipelihara

dalam bentuk *file* dan *output* data dihasilkan sebagai dokumen atau laporan.

b. Perangat Lunak (*Software*)

Perangkat lunak adalah seperangkat instruksi penyimpanan yang memberitahu komputer apa yang harus dilakukan. Komponen ini mencakup tiga tipe dasar perangkat lunak yaitu : perangkat lunak sistem yang mengarahkan pengoperasian perangkat keras, pelaksanaan penerjemahan bahasa dan penyediaan utilities, perangkat lunak aplikasi yang otomatis atau mendukung fungsi bisnis tertentu dan alat-alat komputer, termasuk alat-alat produksi dan pemrograman generasi keempat yang meningkatkan produktivitas atau pengguna dapat mengelola aplikasi mereka sendiri.

c. Perangkat Keras (*Hardware*)

Komputer perangkat keras termasuk semua peralatan fisik yang digunakan untuk *input*, proses, *output*, penyimpanan dan pengiriman data. Perangkat keras diklasifikasikan sebagai satu dari dua bentuk dasar komponen perlengkapannya.

2.3.2.2 Kualitas Informasi

Kualitas informasi yang baik tergantung dari beberapa hal diantaranya yaitu relevan, dapat dipercaya, tepat waktu, akurat dan

ekonomis. Dimana hal-hal tersebut dapat dilihat pada penjelasan berikut :

a. Akurat

Berarti informasi tersebut harus bebas dari kesalahan-kesalahan karena bisa menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

b. Tepat Waktu (*Time Lines*)

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

c. Relevan

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan dinilai kurang relevan dan akan lebih *relevan* bila ditujukan kepada ahli teknik perusahaan.

2.4 Basis Data *(Database)*

Basis data (*Database*) adalah sekumpulan informasi bermanfaat yang diorganisasikan kedalam tata cara yang khusus.

Database adalah kumplan data yang saling berkaitan, berhubungan yang disimpan secara bersama-sama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data-data ini harus mengandung semua informasi untuk mendukung semua kebutuhan sistem. Proses dasar yang dimiliki oleh *database* ada empat, yaitu:

a. Pembuatan data-data baru *(create database)*

b. Penambahan data *(insert)*

c. Mengubah data *(update)*

d. Menghapus data *(delete)*.

Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna. Basis data (*database*) menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan *update* menjadi rumit.

Jadi tujuan dari pengaturan data dengan menggunakan basis data adalah :

a. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.

b. Cara pemasukan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
c. Pengendalian data untuk setiap siklus agar data selalu *up-to-date* dan dapat mencerminakan perubahan spesifik yang terjadi di setiap sistem.
d. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain.

Dalam basis data sistem informasi digambarkan dalam model *entity relationship* (E-R). Bahasa yang digunakan dalam basis data (*database*) yaitu :

a. DDL (*Data Definition Language*)

   Merupakan bahasa definisi data yang digunakan untuk membuat dan mengelola objek *database* seperti *database*, tabel dan *view*
b. DML (*Data Manipulation Language*)

   Merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data pada objek *database* seperti tabel
c. DCL (*Data Control Language*)

   Merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data.

Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data, dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System / DBMS*).

Ada beberapa istilah yang terdapat atau merupakan bagian dari *database* diantaranya yaitu sebagai berikut :

a. *Entity*, adalah sebuah objek yang merupakan bagian dari sistem.
b. *Attribute*, adalah bagian dari *entity* dan memiliki informasi yang dibutuhkan untuk menerangkan informasi yang dikandung suatu *entity*. *Attribute* juga disebut sebagai elemen, *data field*, *data item* untuk mewakili suatu *entity*.
c. *Data value* (nilai atau isi data), adalah data aktual atau informasi yang disimpan pada tiap data elemen atau *attribute*.
d. *Record*, adalah kumpulan elemen-elemen yang saling berkaitan dimana elemen-elemen tersebut menginformasikan tentang suatu *entity* secara lengkap.
e. *File*, merupakan kumpulan *record-record* sejenis yang mempunyai panjang elemen dan *attribute* yang sama namun berbeda-beda data *valuenya*.

2.5 *Database Management System (DBMS)*

"Managemen Sistem Basis Data (Database Management System / DBMS) adalah perangkat lunak yang di desain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar".

Sistem Manajemen Basis data *(Database Management System)* merupakan sistem pengoperasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai. Sistem manajemen *database* merupakan suatu perluasan *software* sebelumnya mengenai *software* pada generasi komputer yang pertama. Dalam hal ini data dan informasi merupakan kesatuan yang

saling berhubungan dan berkerja sama yang terdiri dari: peralatan, tenaga pelaksana dan prosedur data. Sehingga pengolahan data ini membentuk sistem pengolahan data. Peralatan dalam hal ini berupa perangakat keras *(hardware)* yang digunakan, dan prosedur data yaitu berupa perangakat lunak *(software)* yang digunakan dan dipakai untuk mengalokasikan dalam pembuatan sistem informasi pengolahan *database*.

Manipulasi basis data meliputi pembuatan pernyataan (*query*) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (*update*) data, serta pembuatan report dari data. Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi user. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisiensi yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas stuktur data.Sistem manajemen *database* atau *database* management system (DBMS) adalah merupakan suatu sistem software yang memungkinkan seorang user dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. *Database* sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. DBMS yang utuh biasanya terdiri dari :

a. *Hardware*

   Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *databse*. Dalam sebuah organisasi berskala

besar, *hardware* terdiri : jaringan dengan sebuah *server* pusat dan beberapa program *client* yang berjalan di komputer desktop.

b. *Software* beserta *utility Software* adalah DBMS yang aktual. DBMS memungkinkan para *user* untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara *database* dengan *user*. Sebuah *database* harus memuat seluruh data yang diperlukan oleh sebuah organisasi.

c. Prosedur

   Bagian integral dari setiap sistem adalah sekumpulan prosedur yang mengontrol jalannya sistem, yaitu praktik-praktik nyata yang harus diikuti *user* untuk mendapatkan, memasukkan, menjaga, dan mengambil data

d. Data

   Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, adalah kumpulan informasi yang diperlukan oleh suatu organisasi. Jenis data kedua adalah metadata, yaitu informasi mengenai *database*.

e. *User* (Pengguna)

   Ada sejumlah u*ser* yang dapat mengakses atau mengambil data sesuai dengan kebutuhan penggunaan aplikasi-aplikasi dan *interface* yang disediakan oleh DBMS, antara lain adalah :

   1. *Database* administrator adalah orang atau group yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi.

2. *Enduser* adalah orang yang berada di depan *workstatio*n dan berinteraksi secara langsung dengan sistem.
3. *Programmer aplikasi*, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

2.6 *ERD (Entity Relationship Diagram)*

Basis data Relasional adalah kumpulan dari relasi-relasi yang mengandung seluruh informasi berkenaan suatu entitas/ objek yang akan disimpan di dalam *database*. Tiap relasi disimpan sebagai sebuah file tersendiri. Perancangan basis data merupakan suatu kegiaatan yang setidaknya bertujuan sebagai berikut :

a. Menghilangkan redundansi data.
b. Meminimumkan jumlah relasi di dalam basis data.
c. Membuat relasi berada dalam bentuk normal, sehingga dapat meminimumkan permasalahan berkenaan dengan penambahan, pembaharuan dan penghapusan.

ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya.

Contoh : *entity* Mahasiswa, mempunyai atribut nama, umur, alamat, dan nim.

Diagram E-R terdiri dari:

a. Kotak persegi panjang, menggambarkan himpunan entitas.
b. Elips, menggambarkan atribut-atribut entitas.
c. Diamon, menggambarkan hubungan antara himpunan entitas
d. Garis, yang menghubungkan antar objek dalam diagram E-R

E-R Diagram merupakan suatu bahasa pemodelan dimana posisinya dapat dianalogikan dengan *story board* dalam industri film, *blue print* arsitektur suatu bangunan, miniatur, dan lain-lain. Dalam praktiknya, membangun suatu sistem terlebih dahulu dilakukannya suatu perencanaan. Pemodelan merupakan suatu sub bagian dari perencanaan secara keseluruhan sebagai salah satu upaya *feedback* evaluasi perampungan suatu perencanaan. E-R Diagram sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

a. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat
b. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan *developer*.
c. Menghasilkan dokumentasi yang baik untuk *client* sebagai bahan diskusi dengan bentuk E-R Diagram itu sendiri, dan
d. Kamus data bagi bagi para pengembang *database*.

Struktur dari E-R Diagram secara umum ialah terdiri dari:

a. Entitas merupakan objek utama yang informasi akan disimpan, biasanya berupa kata benda, ex; Mahasiswa, Dosen, Nasabah, Mata Kuliah, Ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.
b. Atribut merupakan deskripsi dari objek yang bersangkutan
c. *Relationship* merupakan suatu hubungan yang terjalin antara dua entitas yang ada.

Kardinalitas Relasi ERD yang mempersentasikan suatu basis data relasional senantiasa memiliki relasi-relasi dari sejumlah entitas yang dapat ditentukan banyaknya. Banyaknya suatu relasi yang dimiliki oleh suatu relasi entitas disebut derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas sedangkan derajat minimum disebut dengan modalitas. Kardinalitas yang terjadi diantara dua himpunan entitas (misal A dan B) dapat berupa :

a. *One to One*, satu *record* dipetakan dengan satu *record* di entitas lain. Contoh: satu nasabah punya satu *account*.
b. *One to Many*, Satu *record* dapat dipetakan menjadi beberapa *record* di entitas lain. Contoh: satu nasabah dapat punya lebih dari satu *account*.
c. *Many to Many*, Beberapa *record* dapat dipetakan menjadi beberapa *record* di entitas lain. Contoh: satu nasabah dapat memiliki lebih dari satu *account*. Satu *account* dapat dimiliki lebih dari satu nasabah *(join account).*

2.7 *DFD (Data Flow Diagram)*

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

a. Kebebasan dari menjalankan implementasi teknis sistem.
b. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
c. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
d. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu :

a. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.

b. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
c. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
d. Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

DFD terdiri dari *context diagram* dan diagram rinci *(DFD Levelled)*. *Context diagram* berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *DFD levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Dalam *DFD levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. Jadi dalam *DFD levelled* bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut. Proses yang tidak dapat diturunkan/dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

2.8 Kamus Data (*Data Dictionary*)

Merupakan katalog (tempat penyimpanan) dari elemen-elemen yang berada dalam satu sistem. Kamus data mempunyai fungsi yang sama dalam pemodelan sistem dan juga berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail, dan me-reorganisasi semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

a. Menjelaskan arti aliran data dan penyimpanan dalam DFD
b. Mendeskripsikan komposisi paket data yang bergerak melalui aliran misalnya alamat diuraikan menjadi kota, negara dan kode pos
c. Mendeskripsikan komposisi penyimpanan data
d. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran data.
e. Mendeskripsikan hubungan detail antar penyimpanan yang akan menjadi titik perhatian dalam Diagram Keterhubungan Entitas (E-R)

2.9 MySQL

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya dBase

ataupun Clipper yang masih menggunakan perintah – perintah pemrograman murni.

MySQL adalah sebuah *server database* SQL *multiuser* dan *multi-threaded*. SQL sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program daemon 'mysqld' dan beberapa program lain serta beberapa pustaka. Sebagaimana *database* sistem yang lain, dalam SQL juga dikenal hierarki server dengan database-database. Tiap-tiap *database* memiliki tabel-tabel. Tiap-tiap tabel memiliki field-field. Umumnya informasi tersimpan dalam tabel – tabel yang secara logik merupakan struktur 2 dimensi terdiri atas baris dan kolom. Field-field tersebut dapat berupa data seperti int, realm char, date, time dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada *looping* ataupun percabangan. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal Pascal. MySQL sering digunakan sebagai SQL server karena berbagai kelebihannya, antara lain :

a. Source MySQL dapat diperoleh dengan mudah dan gratis.
b. Sintaksnya lebih mudah dipahami dan tidak rumit.
c. Pengaksesan database dapat dilakukan dengan mudah.

2.10 Delphi

Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan Windows. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah

memanfaatkan metode pemrograman *Object Oriented Programming* (OOP). Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). *Object* biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu.

Khusus untuk pemrograman *database*, Delphi menyediakan *object* yang sangat kuat, canggih, dan lengkap, sehingga memudahkan pemrogram dalam merancang, membuat dan menyelesaikan aplikasi *database* yang diinginkan. Selain itu, Delphi juga dapat menangani data dalam berbagai format *database*, misalnya format MS-Access, SyBase, Oracle, FoxPro, Informix, DB2, MySQL dan lain-lain. Format *database* yang dianggap asli dari Delphi adalah Paradox dan dBase.