BAB II
TINJAUAN PUSTAKA

II.1 Tentang Kalvin Socks Production

Kalvin socks production merupakan perusahaan perorangan yamg bergerak dibidang produksi kaos kaki dan pemasaran dengan menggunakan sistem jaringan pemasaran yang menempatkan 14 agen pemasaran di beberapa lokasi pemasaran dengan tujuan untuk memasok pasar-pasar tradisional yang ada di Indonesia.

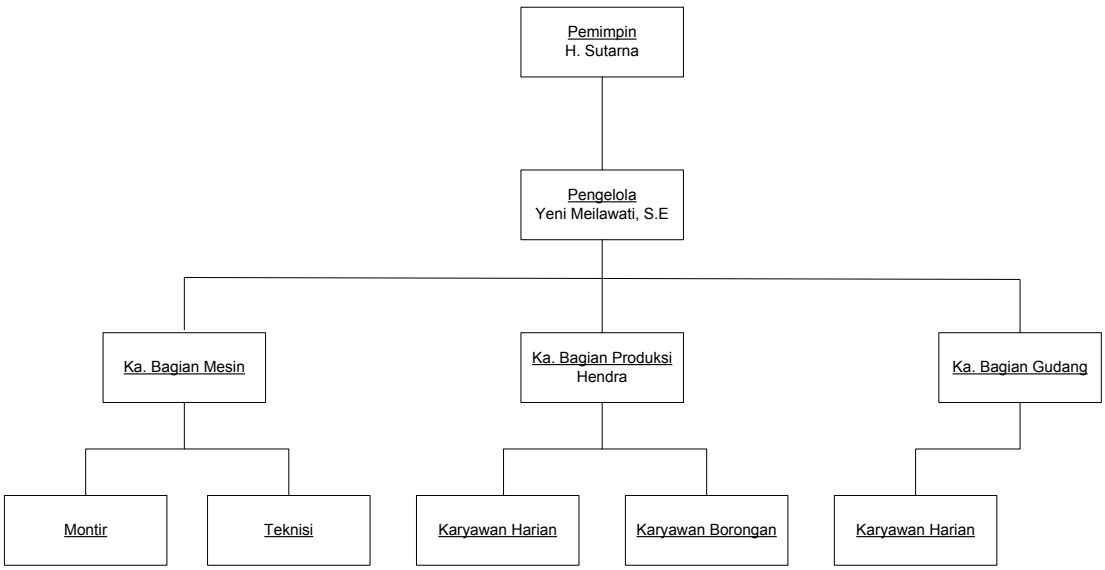
Pada awalnya H. Sutarna sebagai pemilik perusahaan, membangun usaha ini sekitar tahun 1987 dengan bermodalkan semangat dan jiwa usaha yang tinggi untuk meningkatkan taraf hidup dan kesejahteraan keluarga, ketika itu H. Sutarna masih bekerja sebagai karyawan di salah satu perusahaan textile yang ada di Bandung selatan, seiring dengan meningkatnya kebutuhan hidup dan tuntutan hidup yang semakin banyak, maka H. Sutarna berusaha merintis usaha dengan cara berdagang dari pasar ke pasar di waktu liburnya sebagai karyawan, dan setelah H. Sutarna merasa matang dalam berdagang dan mengetahui akan permintaan kaos kaki saat itu, maka H. Sutarna memutuskan memproduksi kaos kaki dengan menggunakan modal awal senilai Rp 1.250.000,00 dari uang pesangon dan 3 unit mesin kaos kaki manual (mesin tangan).

Selain untuk meningkatkan taraf hidup keluarga, usaha ini dibangun juga atas dasar peduli pada keluarga, kerabat, dan lingkungan sekitar. Dengan menciptakan lapangan pekerjaan bagi mereka dan membantu program pemerintah dalam mengurangi tingkat pengangguran, mudah-mudahan usaha ini mendatangkan kebaikan dan memberikan nilai positif bagi kami semua.

Nama Perusahaan	: KALVIN
Lokasi	: Kp. Kebon Kalapa Rt04/Rw03 No. 39 Ds.Batukarut kec. Arjasari Kab. Bandung
Pemilik Perusahaan	: H. Sutarna
Bidang	: Produksi Kaos kaki

II.1.1 Struktur organisasi perusahaan

Struktur organisasi perusahaan adalah gambar bagan yang menjelaskan posisi dan hierarki struktur kerja pegawai di dalam perusahaan. Struktur organisasi perusahaan dari Kalvin Sock Production dapat dilihat pada Gambar II.1.



Gambar II. 1 Struktur Organisasi Kalvin Socks Production

Uraian tugas dan wewenang masing-masing bagian pada struktur organisasi Kalvin Socks Production adalah sebagai berikut :

1. Pengelola :

   Mengawasi kegiatan produksi, penjualan, keuangan, pembelian, administrasi dan kegiatan lain dalam rangka pencapaian tujuan perusahaan

2. Kepala Bagian Mesin

   Mengawasi Seluruh kegiatan produksi yang berhubungan dengan mesin atau alat produksi lainnya.

3. Kepala Bagian Produksi

   Megawasi kegiatan produksi dari awal bahan mentah hingga akhir proses menjadi barang siap jual.

4. Kepala Bagian Gudang

   Mengawasi kegiatan keluar masuk barang yang akan dijual atau dipesan.

II.2 Landasan Teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan dalam melakukan penelitian. Landasan teori yang diuraikan merupakan hasil dari literatur dan buku-buku.

II.2.1 Pengertian data mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*[1].

Menurut Pramudiono, "*Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual."

Menurut Larose, "*Data mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar."

Menurut larose kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa faktor, antara lain :

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetensi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Dari definisi-definisi yang telah disampaikan, hal penting yang terkait dengan *data mining* adalah :

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.

3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Adapun masalah-masalah yang sesuai untuk diselesaikan dengan teknik *data mining* bila dicirikan dengan[2]:

1. Memerlukan keputusan yang bersifat *knowledge-based*.
2. Mempunyai lingkungan yang berubah.
3. Metode yang ada sekarang bersifat sub-optimal.
4. Tersedia data yang bisa diakses, cukup dan relevan.
5. Memberikan keuntungan yang tinggi jika keputusan yang diambil tepat.

II.2.2 Tahapan data mining

Istilah *data mining* dan *knowledge discovery in database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Salah satu tahapan dalam keseluruhan proses KDD adalah *data mining.* Proses KDD secara garis besar dapat dijelaskan sebagai berikut[1]:

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining,* disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-Processing/cleaning*

Sebelum proses data mining dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal. *Pre-processing* data adalah hal yang harus dilakukan dalam proses data mining karena tidak semua data atau atribut data

dalam data digunakan dalam proses data mining. Proses ini dilakukan agar data yang akan digunakan sesuai dengan kebutuhan. Adapun langkah-langkah *preprocessing* adalah sebagai berikut :

a. Pemilihan Atribut (*atribut selection*)

   Pemilihan atribut adalah proses pemilihan mana saja atribut data yang akan digunakan sehingga data tersebut dapat kita olah sesuai dengan kebutuhan proses *data mining.*

b. Pembersihan data (*data cleaning*)

   Proses menghilangkan *noise* dan menghilangkan data yang tidak *relevan* disebut pembersihan data.

3. *Transformation*

Coding adalah proses pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

Data mining adalah mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation.* Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

II.2.3 Arsitektur data mining

Umumnya sistem data mining terdiri dari komponen-komponen berikut:

a. *Database*, data *warehouse*, atau media penyimpanan informasi

Media dalam hal ini bisa jadi berupa *database*, data *warehouse*, *spreadsheets*, atau jenis-jenis penampung informasi lainnya. Data *cleaning* dan data *intregration* dapat dilakukan pada data-data tersebut.

b. *Database* atau data *warehouse server*

Database atau data *warehouse server* bertanggung jawab untuk menyediakan data yang relevan berdasarkan permintaan dari *user* pengguna *data mining*.

c. Basis Pengetahuan *(knowledge base*)

Merupakan basis pengetahuan yang digunakan sebagai panduan dalam pencarian pola.

d. Data mining engine

Yaitu bagian dari *software* yang menjalankan program berdasarkan algoritma yang ada.

e. Pattern evaluation module

Yaitu bagian dari *software* yang berfungsi untuk menemukan *pattern* atau pola-pola yang terdapat di dalam *database* yang diolah sehingga nantinya proses *data mining* dapat menemukan *knowledge* yang sesuai.

f. Graphical user interface

Bagian ini merupakan sarana antara *user* dan sistem *data mining* untuk berkomunikasi, dimana *user* dapat berinteraksi dengan sistem melalui *data mining query*, untuk menyediakan informasi yang dapat membantu dalam pencarian *knowledge*. Lebih jauh lagi, bagian ini mengijinkan *user* untuk melakukan *browsing* pada *database* dan data *warehouse*, mengevaluasi *pattern* yang telah dihasilkan, dan menampilkan *pattern* tersebut dengan tampilan yang berbeda-beda.

II.2.4 Asosiasi (Association)

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisi keranjang belanja. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analiysis* atau *market basket analysis.* Analisis asosiasi atau *Association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item.*

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence. Support* (nilai penunjang) adalah presentase kombinasi *item* tersebut dalam *database*, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar-*item* dalam aturan asosiasi.

Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk *support* (*minimum support*) dan syarat minimum untuk *confidence*(*minimum confidence*).

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap[1] :

1. Analisis pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut.

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \text{..... Persamaan (II.1)}$$

Sementara itu, nilai *support* dari 2 *item* diperoleh dari rumus berikut.

$$Support\ (A,B) = P(A \cap B) \text{..... Persamaan (II.2)}$$

$$Support\ (A,B) = \frac{\Sigma\ Transaksi\ mengandung\ A\ dan\ B}{\Sigma Transaksi} \text{..... Persamaan (II.3)}$$

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang cukup kuat tingkat ketergantungan antar *item* dalam *antecedent* (pendahulu) dan *consequent* (pengikut) serta memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan Asosiatif $A \rightarrow B$.

Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D merepresentasikan himpunan *item* yang berada dalam I. I adalah himpunan *item* yang dijual. Misalkan kita memilih himpunan *item* A dan himpunan *item* lain B, kemudian aturan asosiasi akan berbentuk:

$$jika\ A, maka\ B\ (A \rightarrow B)$$

Dimana *antecedent* A dan *consequent* B merupakan *subset* dari I, dan A dan B merupakan *mutually exclusive* dimana aturan :

$$Jika\ A, maka\ B$$

Tidak berarti

$$Jika\ B, maka\ A$$

Sebuah *itemset* adalah himpunan *item-item* yang ada dalam I, dan *k-itemset* adalah *itemset* yang berisi *k item. Frekuensi itemset* merupakan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan (ϕ). Misalkan ϕ = 2, maka semua *itemset* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent.* Himpunan dari *frequent k-itemset* dilambangkan dengan F_k.

Nilai *confidence dari aturan* $A \rightarrow B$ diperoleh dari rumus berikut.

$$Confidence = P(B \mid A) = \frac{\Sigma\ Transaksi\ mengandung\ A\ dan\ B}{\Sigma\ Transaksi\ mengandung\ A} \quad \text{.....Persamaan(II.4)}$$

II.2.4.1 Langkah-Langkah Proses Aturan Asosiasi

Proses aturan asosiasi terdiri dari beberapa tahap sebagai berikut[4] :

1. Sistem men-*scan database* untuk mendapatkan kandidat 1-*itemset* (himpunan item yang terdiri dari 1 item) dan menghitung nilai *support*nya. Kemudian

nilai *support*nya tersebut dibandingkan dengan *minimum support* yang telah ditentukan, jika nilainya lebih besar atau sama dengan *minimum support* maka *itemset* tersebut dalam *large itemset.*

2. *Itemset* yang tidak termasuk dalam *large itemset* tidak diikutkan dalam itersi selanjutnya(di *prune*).
3. Pada iterasi kedua sistem akan menggunakan hasil *large itemset* pada iterasi pertama(L_1) untuk membentuk kandidat *itemset* kedua(L_2). Pada itersi selanjutnya akan menggunakan hasil *large itemset* pada iterasi sebelumnya(L_{k-1}) untuk membentuk kandidat *itemset* berikut(L_K). Sistem akan menggabungkan(*join*) (L_{k-1}) dengan (L_{k-1}) untuk mendapatkan (L_k), seperti pada iterasi sebelumnya sistem akan menghapus(*prune*) kombinasi *itemset* yang tidak termasuk dalam *large itemset.*
4. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru hasil proses *join* tersebut dihitung *support*nya.
5. Proses pembentuk kandidat yang terdiri dari proses *join* dan *prune* akan terus dilakukan hingga himpunan kandidat *itemset*nya *null,* atau sudah tidak ada lagi kandidat yang akan dibentuk.
6. Setekah itu, dari hasil *frequent itemset* tersebut dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.
7. Pada pembentukan *association rule*, nilai yang sama dianggap sebagai satu nilai.
8. *Association rule* yang berbentuk harus memenuhi nilai minimum yang telah ditentukan.
9. Untuk setiap *large itemset* L, kita cari himpunan bagian L yang tidak kosong. Untuk setiap himpunan bagian tersebut, dihasilkan *rule* dengan bentuk $aB_{(L-a)}$ jika *support*nya (L) dan *support*nya (a) lebih besar dari minimum *support.*

II.2.5 Algoritma Apriori

Ada beberapa algortima yang sudah dikembangkan mengenai aturan asosiasi, namun ada satu algortima klasik yang sering digunakan yaitu algoritma apriori. *Algoritma a priori* termasuk jenis aturan asosiasi pada *data mining.* Ide

dasar dari algoritma ini ialah dimulai dengan mengembangkan *frequent item set* dengan menggunakan satu item dan secara rekrusif mengembangkan *frequent item* set dengan dua item, tiga *item* dan seterusnyaa hingga *frequent item* set dengan semua ukuran. Untuk mengembangkan *frequent item set* dengan satu *item* relatif mudah dilakukan dengan menghitung untuk setiap item, berapa banyak transakasi yang mengandung item tersebut. Jumlah transaksi yang didapat adalah support untuk *set* satu *item* tersebut. Selanjutnya kita bisa menghilangkan *set* satu *item* yang nilai *support*nya dibawah batas tertentu yang kita tetapkan untuk mendapatkan daftar *frequent set* dengan satu *item.*

Untuk mengembangkan *frequent set* dengan dua *item,* kita bisa menggunakan *frequent set* satu item. Alasannya adalah bila *set* satu *item* tidak melebihi *support minimum,* maka sembarang ukuran *item set* yang lebih besar tidak akan melebihi *support minimum* tersebut. Secara umum, mengembangkan *set* dengan F_k – *item* menggunakan *frequent set* dengan *k-1 item* yang dikembangkan dalam langkah sebelumnya. Setiap langkah memerlukan sekali pemeriksaan ke seluruh isi *database*, karena itu algoritma apriori sangat cepat bahkan untuk *database* dengan *item-item* unik dalam jumlah besar.

Langkah-langkah proses pengerjaan algoritma apriori adalah sebagai berikut :

1. Tentukan nilai *minimum support*.
2. Tentukan nilai *minimum confidence*.
3. Cari data kandidat *itemset* ke-k (Ck) dan hitung *frequent itemset*-nya.
4. Tentukan nilai *support*-nya.
5. Hilangkan data yang nilai *support*-nya kurang dari nilai *minimum support*.
6. Setelah mendapatkan data yang nilai *support*-nya lebih dari nilai *minimum support* (*frequent*), gabungkan data-data tersebut sehingga menciptakan k-*itemset*.
7. Ulangi langkah 3 sampai dengan langkah 6 sampai proses penggabungan data tidak menghasilkan data baru.
8. Cari aturan asosiatifnya dengan cara menentukan nilai *confidence*-nya.

9. Hilangkan data yang nilai *confidence*-nya kurang dari nilai *minimum confidence*.

Metode Apriori yang akan digunakan pada penelitian ini, mempunyai beberapa kelebihan :

a. Menggunakan pendekatan apriori untuk mencari maksimal *frequent itemset*.
b. Membutuhkan hanya sedikit pembacaan database dan meminimalkan pengunaan I/O

Faktor-faktor yang dapat mengakibatkan kompleksitas pada algoritma apriori adalah sebagai berikut :

1. Pemilihan *minimum support*
   a. Dengan menurunkan batas *minimum support* dapat menyebabkan semakin banyaknya *frequent itemset* yang didapatkan.
   b. Hal ini juga menyebabkan peningkatan jumlah kandidat dan panjang maksimum dari *frequent itemset*.
2. Dimensi atau jumlah *item* pada *data set*
   a. Lebih banyak ruang yang dibutuhkan untuk menyimpan hitungan *support* untuk setiap item.
   b. Jika jumlah pada *frequent item* juga meningkat, baik komputasi dan I/O cost mungkin juga akan meningkat.
3. Besarnya ukuran database
   a. Algoritma akan meningkat dengan jumlah dari transaksi

II.2.6 Basis Data (*Database*)

Basis Data terdiri dari atas dua kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya

yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya[5].

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:

1. Pembuatan data-data baru *(create database)*
2. Penambahan data *(insert)*
3. Mengubah data *(update)*
4. 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*).

II.2.7 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"[5].

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 :

1. *Hardware*

   Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *database*. Dalam sebuah organisasi berskala besar, *hardware* terdiri : jaringan dengan sebuah server pusat dan beberapa program *client* yang berjalan di komputer desktop.

2. *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.

3. 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.

4. 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*.

5. *User* (Pengguna)

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

a. *Database* administrator adalah orang atau group yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi.
b. End*user* adalah orang yang berada di depan workstation dan berinteraksi secara langsung dengan sistem.
c. Programmer aplikasi, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

II.2.8 *Entity Realtionship Diagram*(ERD)

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak[6]. 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, iasanya 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*).

II.2.9 *Data Dlow Diagram*(DFD)

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.
b. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
c. 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 komp*one*n 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.

II.2.10 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[6].

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).

II.2.11 MySQL

MySQL adalah sebuah program *database* server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*structure Query Language*). MySQL merupakan sebuah *database* server yang free, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lesensinya[7].

MySQL sendiri adalah sebuah *database* server yang mampu menangani beberapa *user* didalamnya. Dengan demikian, MySQL juga mampu menangani beberapa instruksi sekaligus dalam setiap waktu akses. Di dalam sistemnya, MySQL merekam semua data *user* dalam sebuah tabel *user* yang berada pada *database* yang bernama *mysql*. Dalam tabel *user* tersebut demua akses dan hak akses *user* mampu ditangani dengan baik.

Adapun beberapa kelebihan MySQL dalam teknik mengakses data maupun mengelola servernya :

1. *Local Client*

   MySQL sebagai sebuah Server *database*, juga dapat berfungsi sebagai *client* yang dijalankan pada komputer lokal dimana MySQL itu berjalan.

2. *Remote Client*

   Dapat diakses melalui komputer jaringan dengan cara *remote.*

3. *Remote Login*

   Dapat diakses dengan menggunakan program MySQL yang berada pada komputer lain.

4. *Web Browser*

   Dapat diakses menggunakan program yang dibuat dengan sebuah program yang berbasis *server site* yang berjalan di bawah Web *Browser.*

5. *Scripting Language*

   Dapat diakses melalui programaplikasi *client* yang kita ciptakan sendiri dengan menggunakan pemrograman visual maupun non visual yang berjalan di dalam jaringan.

II.2.12 Sekilas Tentang Microsoft Visual Studio .Net Dan C#

Visual studio .Net adalah sekumpulan alat pengembangan *software* yang diperuntukkan bagi .Net platform[8]. Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework).

C# sering dianggap sebagai bahasa penerus C++ atau versi canggih dari C++. C# adalah sebuah bahasa pemrograman yang dikembangkan oleh Microsoft dan menjadi salah satu bahasa pemrograman yang mendukung .Net programming melalui Visual Studio. C# didasarkan pada bahasa pemrograman C++, C# juga

memiliki kemiripan dengan beberapa bahasa pemrograman seperti Visual Basic, Java, Delphi dan tentu saja C++.