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.

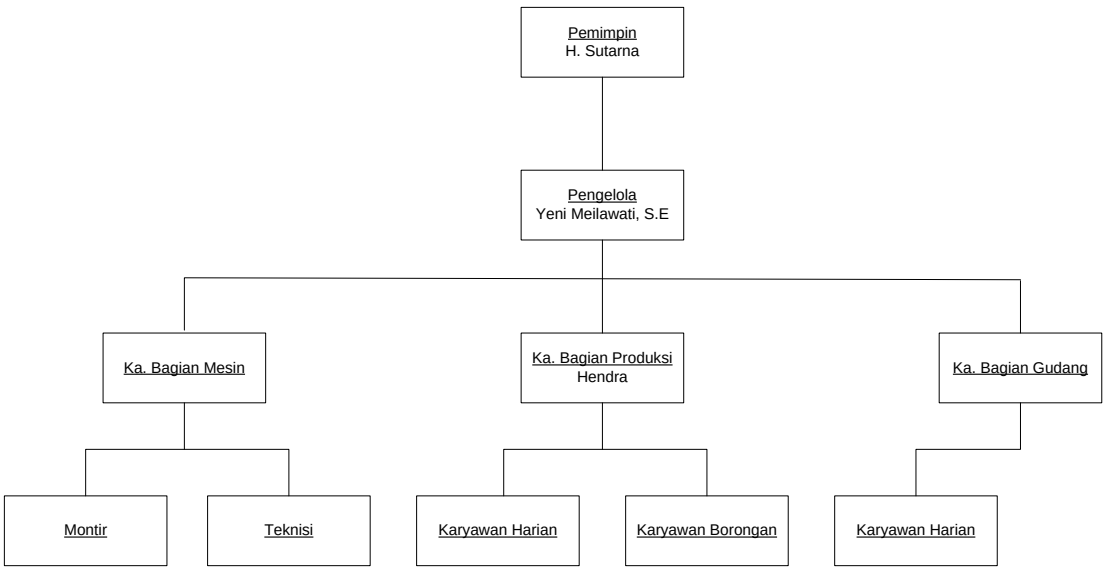
Pada awalnya H. Sutarna sebagai pemilik perusahaan, membangun usaha ini sekitar tahun 1987 dengan bermodalkan semangat dan jiwa usaha yang tinggi untuk meningkatkan taraf hidup dan kesejahteraan keluarga, ketika itu H. Sutarna masih bekerja sebagai karyawan di salah satu perusahaan textile yang ada di 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

III.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 proses menemukan pengetahuan yang menarik dari sejumlah data yang besar yang disimpan di dalam *database*, gudang data, atau repositori informasi lainnya" [6].

Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa f*user*, antara lain [6]:

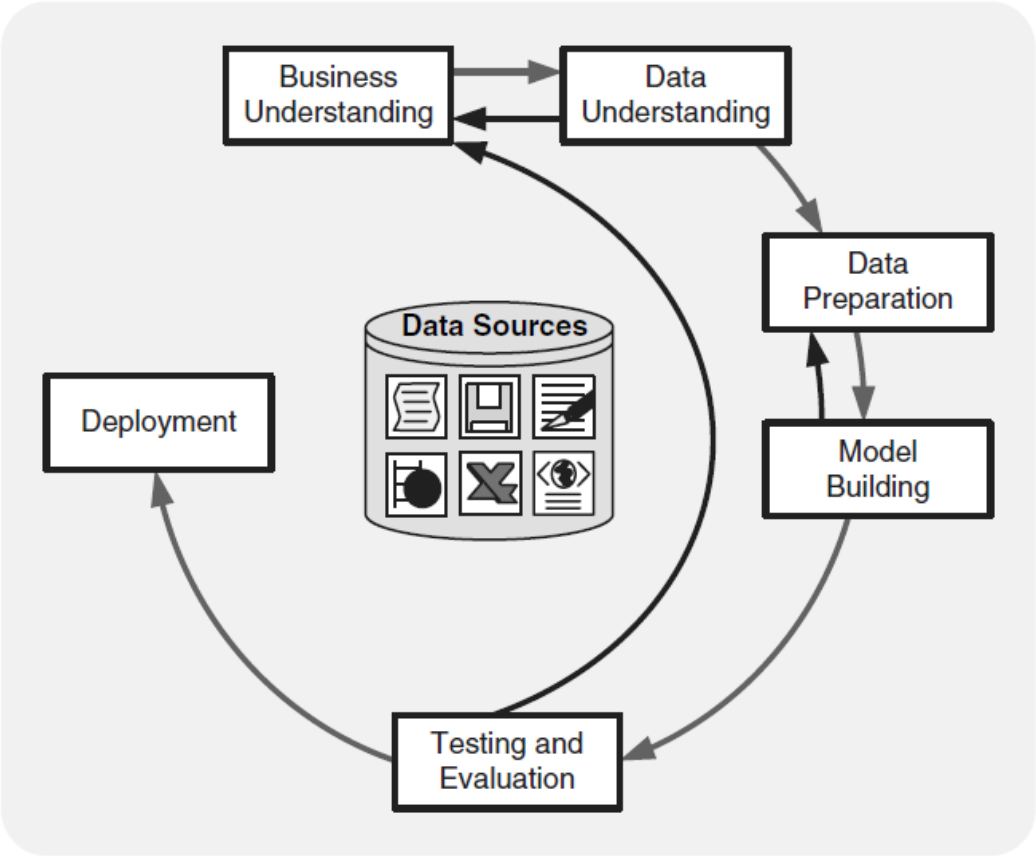
1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Adanya peningkatan akses data melalui *navigasi web* dan *intranet*.
4. Tekanan kompetensi bisnis untuk meningkatkan penguasaan pasar 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.

II.2.2 *Cross- Industry Standard Process for Data Mining* (CRISP- DM)

Cross- Industry Standard Process for Data Mining (CRISP- DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industry seperti Daimler Chrysler, SPSS, dan NCR. CRISP DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP- DM, sebuah proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antarfase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase *modelling*. Berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase *data preparation* untuku perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase *evaluation* [5].



Gambar II. 2 *CRISP- DM* [5]

Enam fase CRISP- DM [5] :

1. Fase Pemahaman Bisnis (*Business Understanding Phase*)
    a. Penentuan tujuan objek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
    b. Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*.
    c. Menyiapkan strategi awal untuk mencapai tujuan
2. Fase Pemahaman Data (*Data Understanding Phase*)
    a. Mengumpulkan data.
    b. Menggunakan analisis penyelidikan data untuk mengenali lebih lanjut data dan pencarian pengetahuan awal.
    c. Mengevaluasi kualitas data
    d. Jika diinginkan, pilih sebagian kecil grup data yang mungkin mengandung pola dari permasalahan
3. Fase Pengolahan Data (*Data Preparation Phase*)
    a. Siapkan dari data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan berat yang perlu dilaksanakan secara intensif.
    b. Pilih kasus dan variabel yang ingin dianalisis dan yang sesuai analisis yang akan dilakukan.
    c. Lakukan perubahan pada beberapa variabel jika dibutuhkan.
    d. Siapkan data awal sehingga siap untuk perangkat pemodelan.
4. Fase Pemodelan (*Modelling Phase*)
    a. Pilih dan aplikasikan teknik pemodelan yang sesuai.
    b. Kalibrasi aturan model untuk mengoptimalkan hasil.
    c. Perlu diperhatikan bahwa beberapa teknik mungkin untuk digunakan pada permasalahan *data mining* yang sama.
    d. Jika diperlukan, proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu.

5. Fase Evaluasi (*Evaluation Phase*)
    a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan.
    b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
    c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
    d. Mengambil keputusan berkaitan dengan penggunaan hasil dari *data mining.*
6. Fase Penyebaran (*Deployment Phase*)
    a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
    b. Contoh sederhana penyebaran : Pembuatan laporan.
    c. Contoh kompleks penyebaran : Penerapan proses *data mining*
    d. secara paralel pada departemen lain**.**

II.2.3 *Knowledge Discovery in Database*

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 [6]:

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.2.4 *Arsitektur Data Mining*

Umumnya sistem *data mining* terdiri dari komponen-komponen berikut[6]:

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.2.5 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* [6].

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 [5] :

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.

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \mathrm{x100}$$.. Persamaan (II.1)

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

$$Support\ (A, B) = P(A \cap B)$$….. Persamaan (II.2)

$$Support\ (A,B) = \frac{\Sigma\ Transaksi\ mengandung\ A\ dan\ B}{\Sigma Transaksi} x100 \ldots..$$ 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.

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

II.2. 5.1 Langkah-Langkah Proses Aturan Asosiasi

Proses aturan asosiasi terdiri dari beberapa tahap sebagai berikut [6]:

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.2.6 Algoritma *Improved Apriori*

Algoritma *Improved Apriori* berbasis *Matrix* diusulkan oleh beberapa peneliti, namun dengan teknik yang berbeda saat pencarian *frequent itemset*. merepresentasikan *database* ke dalam bentuk *matrix* untuk menggambarkan relasi dalam *database*. Kemudian *matrix* dihitung untuk mencari nilai *support* dari *candidate frequent itemset* yang memenuhi kriteria untuk menghasilkan *frequent itemset* tanpa melakukan *scanning* ulang terhadap *database* dengan menggunakan operasi "*AND*" terhadap baris *matrix* sesuai dengan *item* dalam *candidate frequent itemset* dan menambahkan hasil dari *AND*, dengan hasilnya adalah *Support*. [7]

Karena algoritma tidak melakukan *scan* ulang terhadap *database* untuk mencari hubungan seperti algoritma sebelumnya, maka waktu komputasi dan pencarian *candidate frequent itemset* menjadi lebih cepat.

Tahapan algoritma ini berjalan sebagai berikut [2]:

1. *Konversi database* ke dalam bentuk *matrix*.
   a *Konversi database* yang berisi I_n item dan transaksi T_m ke dalam bentuk *matrix*. Baris dari *matrix* mewakili transaksi dan kolom dari *matrix* mewakili *item*. Jika pada suatu transaksi terdapat item maka nilainya adalah 1 dan bernilai 0 jika sebaliknya.
   b Jumlah nilai dari kolom adalah nilai *support count* dan jumlah nilai dari baris adalah banyaknya item dalam suaatu transaksi atau disebut *count*.
2. Periksa jumlah kolom dan jumlah baris
   a Hapus kolom yang jumlah kolomnya kurang dari nilai *minimum support*
   b Hapus baris yang jumlah barisnya kurang dari sama dengan nilai k (*k-frequent itemset)*
3. Gabungkan tiap kolomnya untuk menemukan *frequent 2-itemset* dan gunakan operasi DAN untuk mendapatkan nilainya.
4. Periksa jumlah kolom dan jumlah baris
   a Hapus kolom yang jumlah kolomnya kurang dari nilai *minimum support*.

b Hapus baris yang jumlah barisnya kurang dari sama dengan nilai k (*k-frequent itemset).*

5. Demikian pula untuk mencari K_{th} *-frequent itemset*. Gabungkan tiap kolomnya dan hapus kolom yang kurang dari minimum support dan hapus baris yang jumlah barisnya kurang dari sama dengan k.

II.2.7 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[4].

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.2.8 *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"[4].

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. *Enduser* 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.2.9 *Object Oriented Analysis Design*

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [8].

a. ***Unified Modelling Language***

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal, Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka[8]. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah :

***1. Use Case* Diagram**

Use case adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case* diagram memperlihatkan hubungan diantara *actor* dan *use case*. *User* mempresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang menggambarkan interaksi antara *user* dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk diuji kelayakan sistem[8].

2. Sequence Diagram

Sequence diagram menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior, Sebuah *Sequence diagram* secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam *use case*. *Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang

dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu[8].

3. ***Class* Diagram**

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. *Atribut* merupakan variable-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris *teks* di dalam kotak kelas tersebut. Operasi atau metode adalah fungsi fungsi yang dimiliki oleh suatu kelas[8].

II.2.10 *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 [9].

MySQL adalah sebuah server database SQL *multiuser* dan *multi threaded* [8]. SQL sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program *daemon* 'mysqld'. 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 digunakansebagai SQL server karena berbagai kelebihannya, antara lain:

1. *Source* MySQL dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan *database* dapat dilakukan dengan mudah.

II.2.11 Microsoft Visual Studio .Net Dan *C#*

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*).[10]

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++.