BAB 2
TINJAUAN PUSTAKA

2.1 Profile Tempat Penelitian

PT. PATISA CHAKRA MANDIRI merupakan sebuah perusahaan yang bergerak dibidang air minum dalam kemasan. Perusahaan tersebut merupakan perusahaan yang hanya menjual jasa air kepada berbagai merk. Pada tahun 2005 ide untuk mendirikan industri air minum dalam kemasan ini lebih mendekati pasar. Pada tahun itu juga biaya transport dan kepadatan lalu lintas kiriman semakin meningkat. Seiring bergulirnya waktu pada tahun 2006, kesiapan dana investor ditambah dengan kepastian permintaan kapasitas produksi dari beberapa merk seperti: “Total”, ”Airess”, dan “Aguatik”. Pada tahun 2007 Industri berdiri dengan nama CV. Palasari Tirta Sindanglaya dan mulai beroperasi pada bulan Juli 2007. Pada tahun 2010, perusahaan berganti kepemilikan, dan masuk ke dalam Kabepe Chakra Group, hingga kemudian berganti nama menjadi PT. PATISA CHAKRA MANDIRI.

2.1.1 Logo Perusahaan

Logo dari PT. PATISA CHAKRA MANDIRI dapat dilihat pada gambar 2.1 dibawah ini:



Gambar 2.1 Logo Perusahaan PT. PATISA CHAKRA MANDIRI

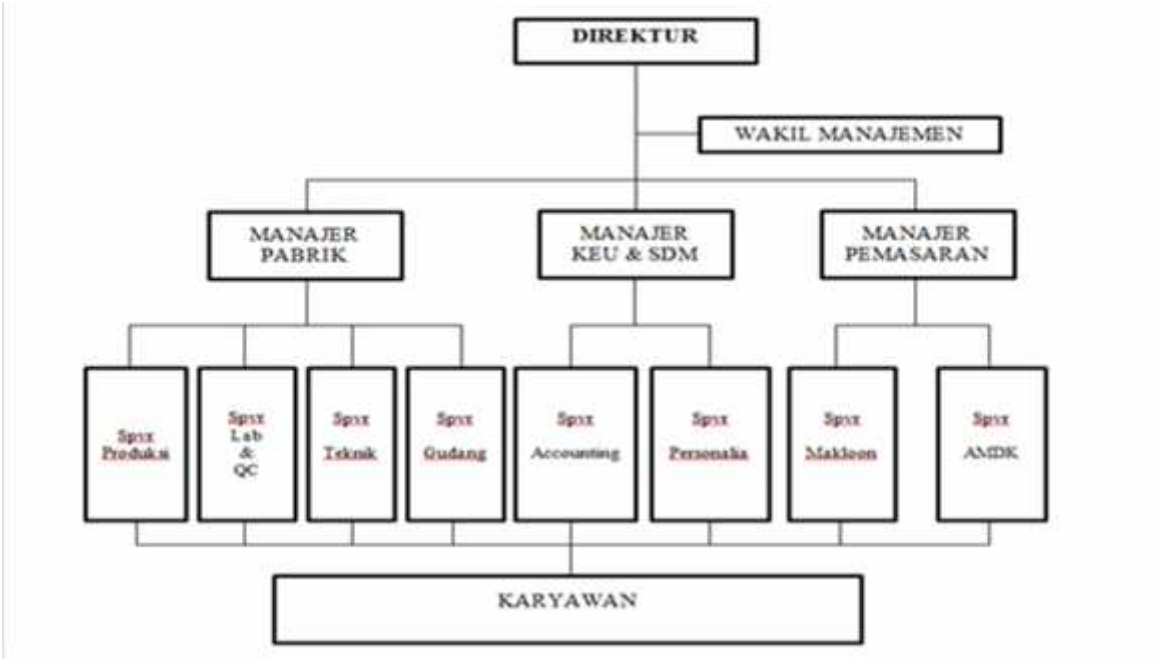
1.1.2 Visi dan Misi Perusahaan

Visi PT. PATISA CHAKRA MANDIRI adalah memberikan warna baru dalam industri kreatif dan menjadi perusahaan air minum dalam kemasan terkemuka di Indonesia. Semua ini disajikan dan terintegrasi dalam keinginan untuk memberikan solusi bagi kebutuhan masyarakat.

PT. PATISA CHAKRA MANDIRI mempunyai misi untuk memberikan produk-produk terbaik dengan sentuhan ide-ide kreatif agar dapat memenuhi kebutuhan di bidang air minum dalam kemasan yang semakin lama semakin berkembang.

1.1.3 Struktur Organisasi Perusahaan

Berikut adalah struktur organisasi dari PT. PATISA CHAKRA MANDIRI, dapat dilihat pada gambar 2.2 dibawah ini:



Gambar 2.2 Struktur Organisasi Perusahaan

Uraian tugas dan wewenang masing-masing bagian pada struktur organisasi PT. PATISA CHAKRA MANDIRI adalah sebagai berikut:

1. Manajer Pabrik:
    a. Mengatur dan mengawasi PPC (Petugas Pengambil Contoh).
    b. Mengatur koordinasi produksi dan kiriman.
    c. Memberikan target produksi.
    d. Mengawasi dan mengendalikan penggunaan material.
    e. Melakukan koreksi kepada PPC.
    f. Melaporkan seluruh kegiatan kepada direksi.
    g. Melakukan hubungan dengan pemakloon.

2. Manajer KEU dan SDM:
    a. Memimpin Department Keuangan dan Sumber Daya Manusia.
    b. Mencari dan menyediakan tenaga kerja/karyawan yang sesuai dengan kebutuhan divisi yang ditentukan.
    c. Melaksanakan pengujian/test kepada calon tenaga kerja/karyawan.
    d. Melaksankan pelatihan/*training* untuk karyawan yang baru.

3. Manajer Pemasaran:
    a. Memimpin Department Pemasaran yang fokus terhadap proses penjualan jasa produksi (Maklon) dan penjualan produk AMDK *merk* “Patisa”.
    b. Merencanakan kegiatan pemasaran secara sistematis yaitu efektif dan efisien.
    c. Bertanggung jawab atas ketidaksesuiaan pada target penjualan dan penagihan.

4. Supervisor Produksi:
    a. Melaksanakan produksi sesuai dengan target pengiriman barang.
    b. Melakukan target produksi sesuai dengan target.
    c. Bekerjasama dengan gudang teknisi dalam operasional produksi.

d. Melakukan tindakan administrasi yaitu meng-*input* laporan produksi, kemudian laporan tersebut dibuat grafik.
e. Mengoreksi gudang teknisi dan personil.
f. Menjeaskan dan mengarahkan untuk melaksanakan pekerjaan produksi yang benar kepada pelaksana produksi.
g. Melakukan koordinasi dan kerjasama yang baik dengan bagian lain.

5. Supervisor Laboratorium dan *Quality Control*:
   a. Memimpin bagian teknik sebagai unit kerja yang fokus terhadap kesiapan mesin-mesin dan fasilitas pendukung proses produksi AMDK lainnya.
   b. Merencankan pekerjaan Maintenance (perwatan dan perbaikan) mesin-mesin AMDK secara sistematis yaitu efekif dan efisien.
   c. Mendukung jalannya mesin-mesin yang ada.
   d. Menjelaskan dan mengarahkan untuk melaksanakan pekerjaan teknik yang benar kepada pelaksana/teknisi.
   e. Bertanggung jawab atas ketidaklancaran proses produksi yang diakibatkan oleh oleh kerusakan mesin-mesin.
   f. Bertanggung jawab atas ketidaksesuian kualitas yang disebabkan oleh mesin yang tidak berfungsi dengan baik.

6. Supervisor Teknik:
   a. Memimpin bagian teknik sebagai unit kerja yang fokus terhadap kesiapan mesin-mesin dan fasilitas pendukung proses produksi AMDK lainnya.
   b. Mendukungg jalannya pencapaian proses produksi sesuai target yaitu dengan menjamin kelancaran mesin-mesin yang ada.
   c. Mengarahkan untuk melaksanakan pekerjaan teknik yang benar kepada pelaksana/teknisi.
   d. Membuat laporan hasil kerja dan mengevaluasi secara periodik.

7. Supervisor Gudang:
    a. Memimpin bagian gudang bahan baku (*material*) dan produksi jadi sebagai unit kerja yang fokus terhadap proses penerimaan dan penyerahan *material*.
    b. Merencanakan pekerjaan penerimaan, penyimpanan, penyerahan material.
    c. Menjaga dan merawat material dan atau produk jadi tetap dalam kondisi yang baik sesuai standar yang ditetapkan.
    d. Bertanggung jawab atas ketidaksesuaian hasil *stock opname*.

8. Supervisor *Accounting*:
    a. Memimpin bagian produksi sebagai unit kerja yang fokus terhadap proses pembukuan dalam keuangan dan kegiatan pekerjaan yang berkaitan.
    b. Melaksankan pekerjaan accounting secara sistematis yaitu efektif dan efisien, akuntabilitas dan mutakhir (*up to date*).
    c. Mengarahkan dan melaksanakan pekerjaan pembukuan yang benar kepada pelaksana pembukuan.
    d. Membuat laporan hasil kerja serta melakukan evaluasi secara periodik.
    e. Bertanggung jawab atas ketidaksesuiaan saldo kas.
    f. Bertanggung jawab atas ketidaksesuiaan pada pembukuan keuangan.
    g. Bertanggung jawab kepada Manajer Keuangan dan SDM.

9. Supervisor Personalia:
    a. Mempersiapkan kebutuhan personil untuk produksi.
    b. Melakukan tindakan koreksi baik personil PPC (Petugas Pengambil Contoh) sesuai bagian yang melapor.
    c. Melakukan koordinasi keamanan.
    d. Melakukan kordinasi humas dan limbah.
    e. Melakukan hubungan dengan pihak luar.
    f. Merencanakan pekerjaan ketenagakerjaan secara sistematis yauitu secara efektif dan efisien.

10. Supervisor Pemaklon:
    a. Memimpin bagian produksi sebagai unit kerja yang fokus terhadap proses kerjasama maklon dan kegiatan pekerjaan yang berkaitan.
    b. Melaksanakan kegiatn pekerjaan pemasaran maklon berdasarkan kapasitas mesin terpasang.

11. Supervisor AMDK:
    a. Memimpin bagian pemasaran AMDK sebagai unit kerja yang fokus terhadap penjualan AMDK *merk* "Patisa" an keiatan yang berkaitan.
    b. Melaksanakan kegiatan pemasaran AMDK *merk* "Patisa" berdasarkan target yang ditetapkan.
    c. Bertanggung jawab atas ketidaksesuaian pada target penjualan.
    d. Bertanggnng jawab kepada manajer pemasaran.
    e. Bertanggung jawab atas kepuasan pelanggan.

2.2 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.

2.2.1 Pengertian Data, Informasi dan *Knowledge*

Data merupakan fakta yang dikumpulkan, disimpan, dan diproses oleh sebuah sistem informasi. Saat ini, akumulasi pertumbuhan jumlah data berjalan dengan cepat dalam format dan basis data yang berbeda. Data-data tersebut, antara lain [3]:

1. Data operasional atau transaksional, seperti penjualan, inventaris, penggajian, akuntansi, dan sebagainya.
2. Data nonoperasional, seperti industri penjualan (supermarket), peramalan, dan data ekonomi makro.
3. Metadata, adalah data mengenai data itu sendiri, seperti desain logika basis data atau definisi kamus data.

Sementara informasi adalah pola, asosiasi, atau hubungan antara semua data yang dapat memberikan informasi. Sebagai contoh, analisis titik eceran (retail point) data transaksi penjualan dapat menghasilkan informasi mengenai produk apa yang sebaiknya dijual dan kapan menjualnya.

Pengetahuan (knowledge) sebenarnya merupakan sebuah informasi juga yang merupakan hasil dari pengolahan data. Vercellis (2009) memandang bahwa suatu informasi dikatakan pengetahuan jika dapat digunakan dalam pengambilan keputusan. Dengan demikian pengetahuan dapat dijelaskan kembali sebagai kumpulan dari data dan informasi yang bertemu dengan kompetensi dan pengalaman seseorang untuk menindak lanjuti data dan informasi yang ada sehingga dapat dikembangkan untuk pengambilan suatu keputusan. Tidak seperti informasi yang hanya bersifat memberi tahu, pengetahuan harus mampu digunakan untuk proses pengambilan keputusan. Salah satu tahapan dalam keseluruhan proses KDD adalah data mining. Proses KDD secara garis besar dapat dijelaskan sebagai berikut [4]:

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 (*typography*). 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 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.

2.2.2 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).

2.2.3 *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"[8].

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. **S**oftware 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 bertanggung jawab mengimplementasikan sistem database di dalam suatu organisasi.
   b. Enduser adalah orang yang berada di depan workstation dan berinteraksi secara langsung dengan sistem. Programmer aplikasi, orang yang berinteraksi dengan database melalui cara yang berbeda.

2.2.4 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" [4].

Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa *user*, antara lain [4]:

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.

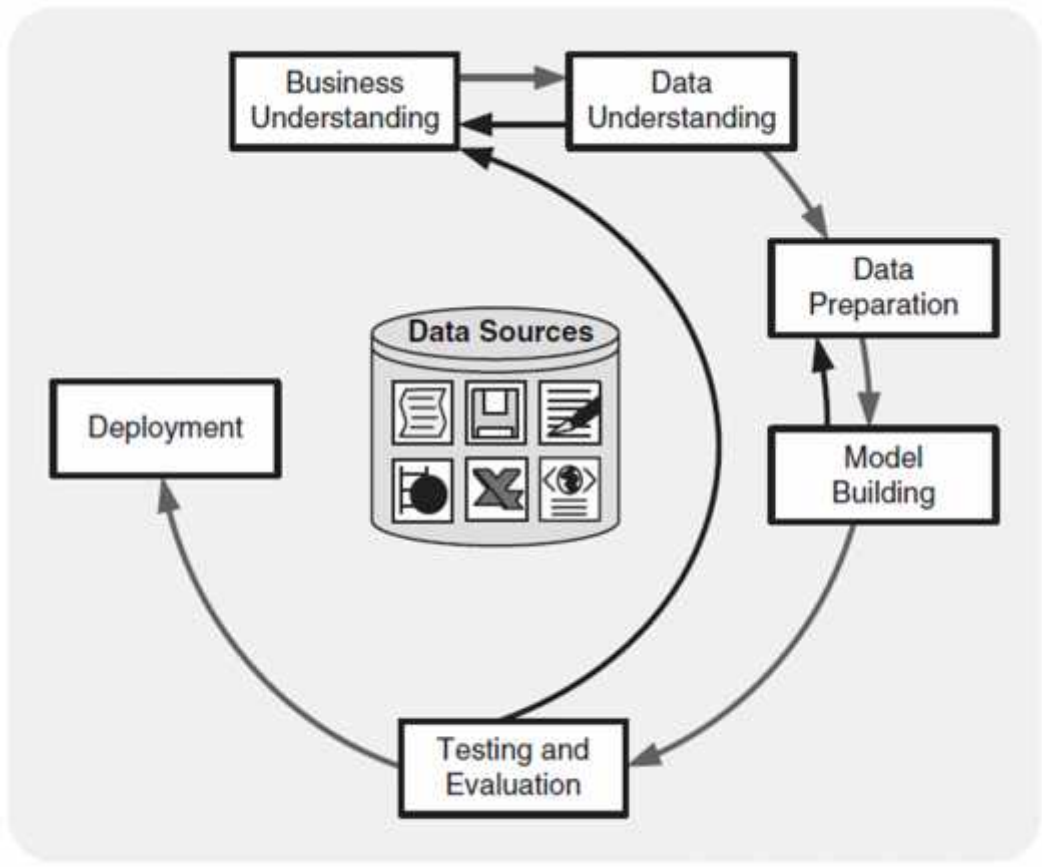
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

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.

2.2.5 *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 2.3 *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**.**

2.2.6 *Arsitektur Data Mining*

Umumnya sistem *data mining* terdiri dari komponen-komponen berikut[4]:

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.

2.2.7 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* [4].

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 [4]:

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.

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \mathrm{x}100 \text{..}$$ Persamaan (2.1)

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

$$Support\ (A, B) = P(A \cap B) \text{…..}$$ Persamaan (2.2)

$$Support\ (A, B) = \frac{\Sigma Transaksi\ mengandung\ A\ dan\ B}{\Sigma Transaksi} \mathrm{x}100 \text{…..}$$ Persamaan (2.3)

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.

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

2.2.7.1 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.*

2.2.8 Algoritma *FP-Growth*

Algoritma yang sama dengan Apriori, FP-Growth dimulai dengan menghitung item tunggal sesuai dengan jumlah kemunculan item yang ada didalam *dataset*. Setelah proses penghitungan selesai maka akan dibuat struktur pohon pada tahap kedua. Pohon yang dibuat mulanya kosong yang nanti akan diisi dengan hasil dari dataset yang telah didapat sebelumnya. Kunci untuk mendapatkan struktur pohon yang bisa didapatkan dengan proses lebih cepat

untuk mencari item set yang besar menjadi sedikit dengan di urutkan secara *descending* dari frekuensi yang ada *dataset* tersebut. Masing-masing item yang tidak mencapai kebutuhan minimum dari *threshold* tidak dimasukkan kedalam pohon, tapi dikeluarkan secara efektif dari *dataset* [6].

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. FP-Tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-Tree. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data FP-Tree semakin efektif. Kelebihan dari FP-Tree adalah hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien.

Adapun FP-Tree adalah sebuah pohon dengan definisi sebagai berikut:

a FP-Tree dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan berupa pohon yang beranggotakan item-item tertentu, dan sebuah tabel *frequent header*.

b. Setiap simpul dalam FP- Tree mengandung tiga informasi penting, yaitu label item, menginformasikan jenis item yang direpresentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi yang melalui simpul tesebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label item sama antar-lintasan, ditandai dengan garis panah putus-putus.

2.2.8.1 Langkah-Langkah Proses Perhitungan Association Rule Dengan Algoritma FP-Growth

Proses perhitungan *association rule* terdiri dari beberapa tahap adalah sebagai berikut [7]:

1. Membuat *Header Item*

   Header dalam hal ini selain sebagai *header* suatu *item* ke FP-Tree juga sebagai jenis *item* dasar yang memenuhi minimum *support*. Setelah

mendapatkan *item* dan nilai *support*-nya, maka *item* yang tidak *frequent* dibuang dan *item* diurutkan berdasarkan nilai *support*-nya. Header untuk item, disiapkan pada suatu *array* tertentu dan ditambahkan ketika membuat FP-Tree.

2. Membuat FP-Tree

   FP-Tree dibangun dengan mencari *item* sesuai urutan pada *item* yang *frequent*. Data transaksi tidak perlu diurutkan, dan untuk tiap *item* yang ditemukan bisa langsung dimasukkan ke dalam FP-Tree. Sesudah membuat *root*, tiap *item* yang ditemukan dimasukkan berdasarkan *path* pada FP-Tree. Jika *item* yang ditemukan sudah ada, maka nilai *support item* tersebut yang ditambahkan. Namun jika *path* belum ada, maka dibuat *node* baru untuk melengkapi *path* baru pada FP-Tree tersebut. Hal ini dilakukan selama item pada transaksi masih ada yang *qualified*, artinya memenuhi nilai minimum *support*. Jadi, *item-item* yang ditemukan dalam transaksi akan berurutan memanjang ke bawah. Dalam struktur FP-Tree, diterapkan alur *path* dari *child* hingga ke *root*. Jadi, suatu *path* utuh dalam FP-Tree adalah dari *child* terbawah hingga ke *root*. Tiap *node* pada FP-Tree memiliki *pointer* ke *parent*, sehingga pencarian harus dimulai dari bawah.

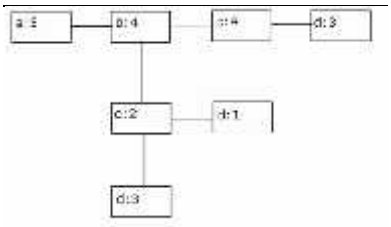
3. *Pattern Extraction*

   Pattern extraction dilakukan berdasarkan keterlibatan *item* pada suatu *path*. Di setiap *path*, diperiksa semua kombinasi yang mungkin dimana *item* tersebut terlibat. Di iterasi berikutnya dilakukan dengan melibatkan *item* berikutnya, tanpa melibatkan *item* sebelumnya, sehingga *pattern* yang sama tidak akan ditemukan dua kali pada *path* yang sama. Bila *item* pertama suatu hasil kombinasi bukan *item* terakhir (sebelum *root*), maka kombinasi *itemset* tersebut masih bisa dikembangkan lagi.

4. Memasukkan setiap pattern yang ditemukan ke dalam *PatternTree*

   Setelah mengolah FP-Tree menjadi *pattern-pattern*, diperlukan proses akumulasi *pattern-pattern* yang ditemukan mengingat *pattern* yang sama

dapat ditemukan pada path yang berbeda. Untuk itu digunakan struktur data *Pattern Tree* (lihat Gambar 2.5). Setiap node di *Pattern Tree* merepresentasikan dan menyimpan frekuensi suatu *pattern*. *Pattern Tree* terdiri atas *Pattern TreeNode* yang menyimpan nilai item, nilai *support* dan dilengkapi dengan dua *pointer* yaitu untuk horisontal dan vertikal.



Gambar 2.5 *Pattern Tree*

Misalnya pada *node* d:1 di atas, berarti terdapat *pattern* a-c-d bernilai *support* 1. Kemudian bila ada pattern a-c-d lagi bernilai *support* n yang ditemukan dari FP-Tree maka nilai *support* 1 tersebut menjadi n+1. Contoh hasil lengkap dari *PatternTree* tersebut:

1. a:5 menggambarkan bahwa ada pattern a sebanyak 5
2. b:4 menggambarkan bahwa ada pattern a-b sebanyak 4
3. c:4 menggambarkan bahwa ada pattern a-b-c sebanyak 4
4. d:3 menggambarkan bahwa ada pattern a-b-c-d sebanyak 3
5. c:2 menggambarkan bahwa ada pattern a-c sebanyak 2
6. d:1 menggambarkan bahwa ada pattern a-c-d sebanyak 1
7. d:3 menggambarkan bahwa ada pattern a-d sebanyak 3

5. Mengurutkan dan Menyeleksi *Pattern*

   Pattern yang tidak memenuhi minimum *support*, dihapus dari daftar *pattern*. *Pattern-pattern* yang tersisa kemudian diurutkan untuk memudahkan pembuatan *rules*.

2.2.9 *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 [9].

1. 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:

***a. 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[9].

b. 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[9].

c. *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[9].

2.2.10 Microsoft Visual Studio

Visual studio .Net adalah sekumpulan alat pengembangan *software* yang diperuntukkan bagi .Net platform[10]. 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++.

2.2.11 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *structured Query Language* (SQL). MySQL dalam operasi *client server* melibatkan *server daemon* MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client.* MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku bahwa MySQL yang mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data [11].

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai bahasa yang standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan DBASE atau *Clipper* yang masih menggunakan perintah-perintah pemrograman.