BAB 2
TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

AALIYAH merupakan butik yang berdiri sejak tahun 2009 di Bandung yang berdiri sejak awal-awal industri butik hijab belum menjamur. Perusahaan yang bergerak di industri *fashion* umum dan hijab ini mengambil nama AALIYAH yang artinya tempat tertinggi, dengan harapan AALIYAH menjadi butik yang berada ditempat tertinggi dihati pelanggannya.

Perusahaan ini mengusung konsep hijab yang *mix and match*, sehingga dengan baju yang dimiliki sebelumnya, wanita yang ingin berhijab bisa dengan mudah mencari baju tambahan atau jilbab di AALIYAH sebagai pelengkap gaya berbusananya. Konsep ini juga menggabungkan penjualan produk fashion umum dan hijab sehingga menciptakan pasar yang luas untuk AALIYAH. Desain produk AALIYAH juga berwarna netral dan berunsur tanah sehingga mudah untuk dipadupadankan dengan baju-baju yang sudah ada. Untuk desain yang digunakaan AALIYAH menggunakan campuran bahan *lace* (bahan bahan feminin yang lembut) dan *jacquard* atau *suede* yang lebih maskulin, menampilkan desain yang lebih kaya dan tidak monoton. Semua ini disajikan dan terintegrasi dalam keinginan untuk memberikan solusi bagi kebutuhan *fashion* wanita.

2.1.1 Logo Perusahaan

Logo dari AALIYAH dapat dilihat pada gambar 2.1 dibawah ini:



Gambar 2. 1 Logo AALIYAH

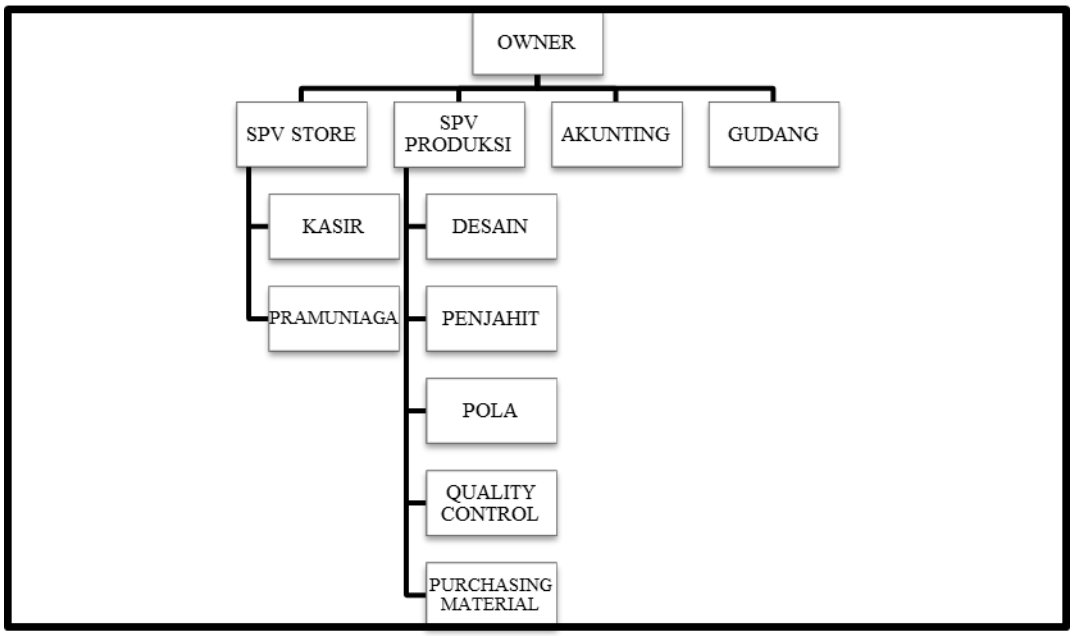
2.1.2 Visi dan Misi Perusahaan

AALIYAH mempunyai visi untuk memberikan desain yang dapat dipadupadankan sehingga memudahkan pengguna hijab pemula serta memberikan solusi bagi kebutuhan berhijab masa kini, dan kedepannya diharapkan dapat menambah pengguna hijab baru.

Misi dari AALIYAH untuk memberikan desain yang *wareable*, tidak monoton dan dengan konsep toko yang menggabungkan antara pakaian hijab dan non-hijab menjadikan cakupan pembelinya lebih luas dan dapat memenuhi kebutuhan dibidang *fashion* yang semakin lama semakin berkembang.

2.1.3 Struktur Organisasi Perusahaan

Berikut adalah struktur organisasi dari AALIYAH:



Gambar 2. 2 Struktur Organisasi Perusahaan

Uraian tugas dan wewenang masing-masing bagian pada struktur organisasi Aaliyah adalah sebagai berikut:

1. *Owner*:
    a. Mengambil keputusan setiap kegiatan di AALIYAH.
    b. Menerima laporan penjualan, laporan persediaan barang dan laporan barang masuk.
2. *Spv Store*:
    a. Melakukan *checklist day to day*.
    b. Verifikasi surat permintaan barang.
    c. Memonitor penataan barang.
    d. Mengawasi kelancaran transaksi konsumen di kasir.
3. *Spv* Produksi:
    a. Mengawasi kegiatan produksi dari awal desain, kain yang dipilih, proses pola serta baju yang dijahit.
    b. Pengecekan penerimaan barang secara kualitas dan kuantitas.
4. Akunting:
    a. Mengatur arus kas kecil dan kas besar di AALIYAH.
    b. Membuat laporan keuangan tiap bulan dan tahun.
    c. Mengurus pembayaran gaji karyawan.
5. Gudang:
    a. Mengecek jumlah stok barang di gudang.
    b. Membuat laporan persediaan barang.
    c. Membuat laporan barang masuk.
    d. Membuat laporan barang keluar.

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 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[3].

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 ini harus mengandung semua informasi untuk mendukung semua kebutuhan sistem. Proses dasar yang dimiliki oleh *database* ada empat, yaitu:

1. Pembuatan 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*).

2.2.2 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"[3].

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 *owner*. 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 *owner* 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*

Software beserta utility *Software* adalah DBMS yang aktual. DBMS memungkinkan para *owner* untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara *database* dengan *owner*. 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 *owner* 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. *Endowner* 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.

2.2.3 *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]:

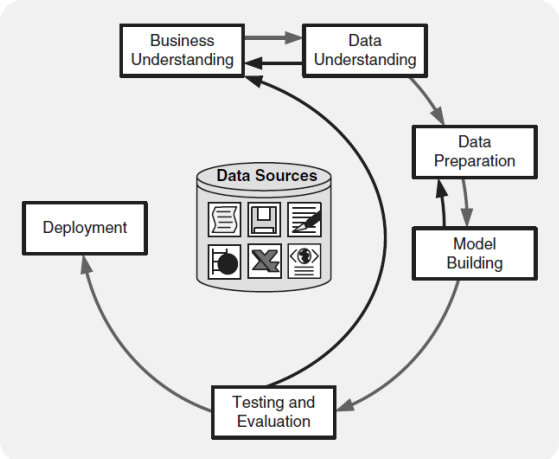
1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Adanya peningkatan akses data melalui *navigasi web* dan *intranet*.
4. Tekanan kompetensi bisnis untuk meningkatkan penguasaan pasar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

2.2.3.1 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* [2].



Gambar 2. 3 CRISP-DM

Enam fase CRISP- DM [2]:

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**.**

2.2.3.2 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 [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*

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

a. Pembersihan data (*data cleaning*)

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

b. 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*.

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 d*item*ukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.2.4 *Association Rule*

Association rule merupakan sutu proses pada *data mining* untuk menentukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support* (*minsup*) dan *confidence* (*minconf*) pada sebuah *database*. Kedua syarat tersebut akan digunakan untuk *interesting association rules* dengan dibandingkaan dengan batasan yang telah ditentukan, yaitu *minsup* dan *minconf*. [7]

Association rule mining adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *dataset*. Dimulai dengan mencari *frequent itemset*, yaitu kombinasi yang paling sering terjadi dalam suatu *itemset* dan harus memenuhi *minsup*. [7]

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. Pencarian *Frequent Itemset*

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

$$Support(A) = \frac{Jumlah\ transaksi\ yang\ mengandung\ A}{Total\ Transaksi}$$

Kemudian, untuk mendapatkan nilai *support* dari dua *item* dapat diperoleh dengan rumus berikut:

$$Support\ (A,B) = \frac{Jumlah\ transaksi\ yang\ mengandung\ A\ dan\ B}{Total\ Transaksi}$$

Setelah semua *frequent item* dan *large itemset* didapatkan, dapat dicari syarat minimum confidence (*minconf*) dengan menggunakan rumus berikut:

$$Confidence\ (A,B) = \frac{Jumlah\ Transaksi\ mengandung\ A\ dan\ B}{Jumlah\ Transaksi\ mengandung\ A} \text{ x } 100\%$$

2. *Generate Rule*

Setelah *frequent itemset* 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 $\phi = 2$, 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.

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

2.2.5 Algoritma FP-Growth

Algoritma FP-Growth merupakan pengembangan dari algoritma *Apriori*. Algoritma *Frequent Pattern Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. [5]

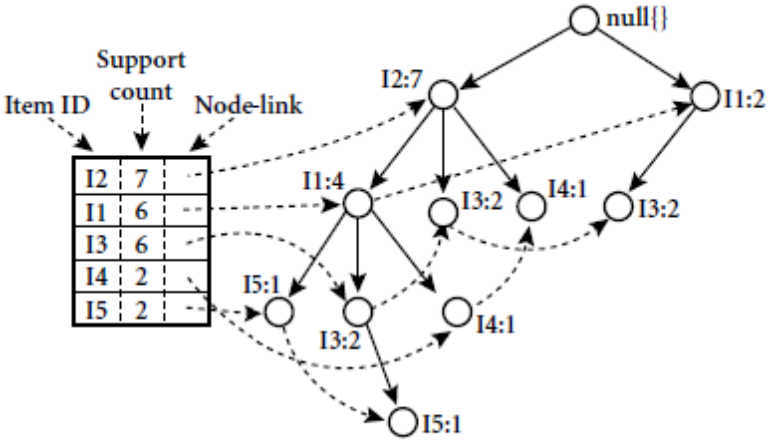
Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree*, yang biasa disebut *FP-Tree*, dalam pencarian *frequent itemset* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma *Apriori*. Dengan menggunakan konsep tersebut, algoritma *FP-Growth* menjadi lebih cepat dari algoritma *Apriori*. [6]

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.

Metode *FP-Growth* dibagi menjadi tiga tahapan, yaitu [4]:

1. Tahap pembangkitan *conditional pattern base.*
2. Tahap pembangkitan *conditional FP-Tree.*
3. Tahap pencarian *frequent patterns generated.*



Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Gambar 2. 4 Tahap Pembentukan *FP-Tree*

2.2.6 *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 d*item*ui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [8].

1. Unified Modelling Language

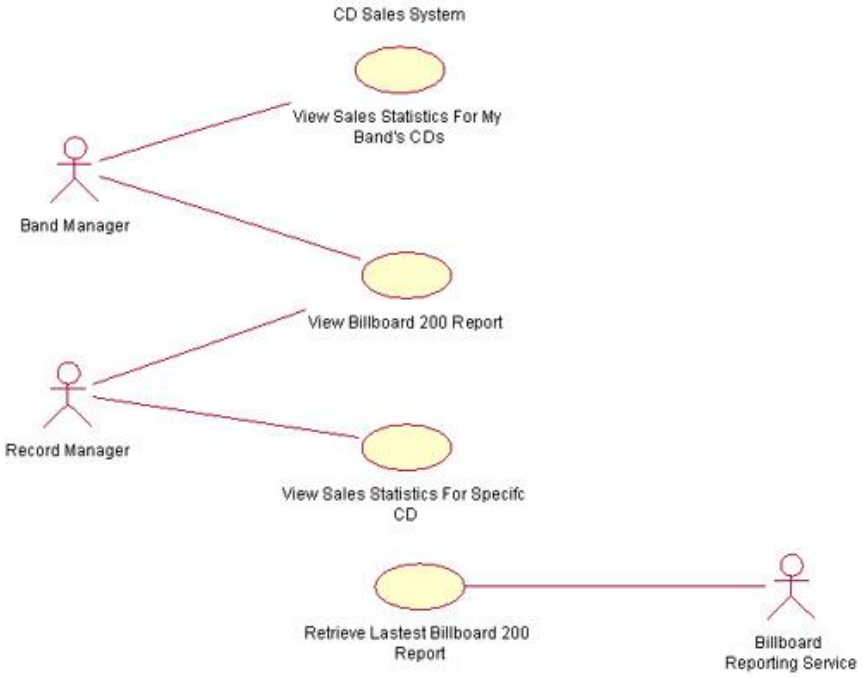
UML dimaksudkan untuk menjadi sebuah bahasa pemersatu yang memungkinkan para professional IT membuat model aplikasi. Penulis utamanya adalah Jim Rumbaugh, Ivar Jacobson dan Grady Booch yang awalnya mereka memiliki metode sendiri (OMT, OOSE dan Booch). Mereka bergabung dan membawa standar yang bebas dan terbuka (fenomena yang menghasilkan J2EE, SOAP dan Linux). Salah satu alasan yang telah menjadi UML standar bahasa

pemodelan adalah bahasa *programming* yang independen. Alat modeling UML berasal dari IBM Rational yang digunakan pada J2EE *shops* yang sama pada .Net *shops*. Notasi UML yang ditetapkan adalah bahasanya bukan metodologinya, hal ini penting karena bahasa adalah lawan dari metodologi yang dapat dengan mudah masuk ke setiap perusahaan-perusahaan bisnis tanpa melakukan perubahan.

UML bukan merupakan metodologi sehingga tidak memerlukan produk kerja formal (seperti artefak IBM *Rational Unified Process Lingo*). UML memberikan beberapa jenis diagram yang diguankan dalam metodologi tertentu, meningkatkan kemudahan dan memahami dalam pengembangan pemodelan aplikasi.[9]

a. *Use Case Diagram*

Use case diagram adalah sebuah diagram yang bertujuan untuk mengkomunikasikan proses bisnis yang terjadi dan ruang lingkup dari sebuah sistem. *Use case diagram* terdiri dari aktor dan proses bisnis yang mereka lakukan melalui sistem tersebut. Contoh use case diagram:

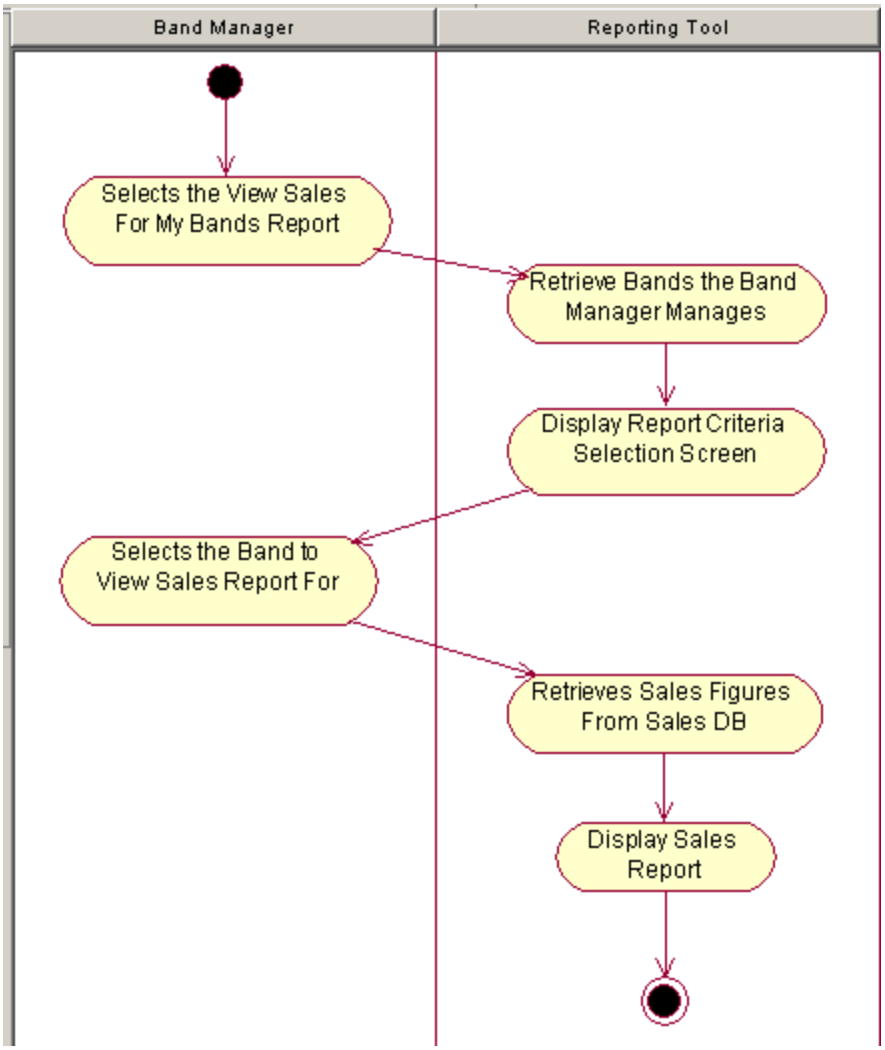


Gambar 2. 5 Contoh *Use Case Diagram*

Gambar 2.5 menunjukkan contoh sistem penjualan. Sistem ini memungkinkan para *manager band* melihat penjual dan laporan statistik *Billboard* 200.

b. *Activity Diagram*

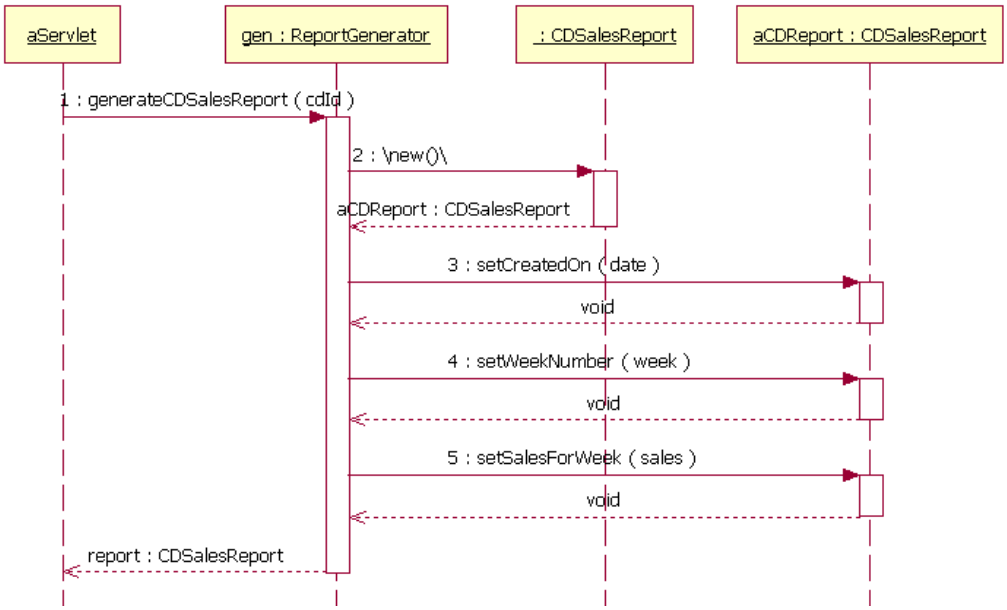
Activity diagram adalah sebuah diagram untuk memodelkan perilaku use cases dan objects di dalam system. Diagram ini menggambarkan bagaimana hubungan antara *use case* dan obyek yang terdapat di dalam sistem, dari suatu keadaan awal sampai keadaan akhir. Contoh yang ada di bawah adalah salah satu contoh *activity diagram* menggunakan *swim lane*, dimana kegiatan- kegiatan yang berlangsung di dalam suatu *use case* dipisahkan menurut aktor yang melakukannya.



Gambar 2. 6 Contoh *Activity Diagram*

c. *Sequence Diagram*

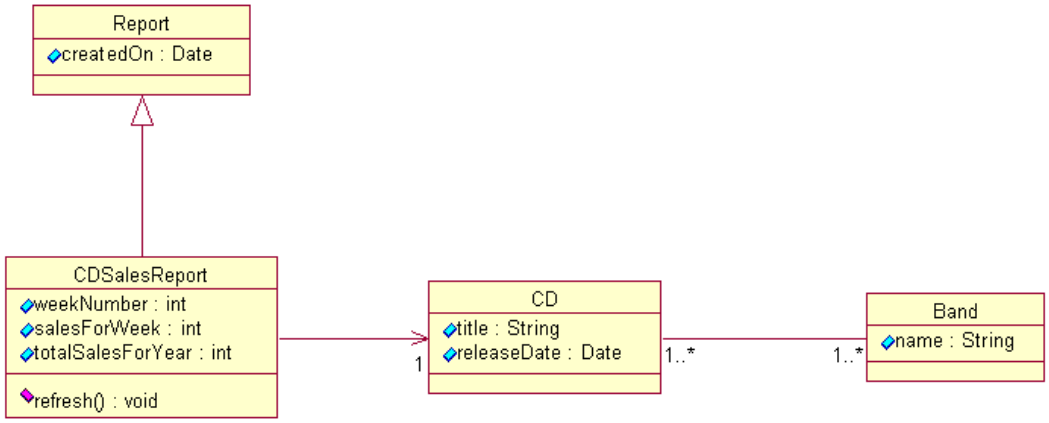
Sequence diagram adalah sebuah diagram untuk memodelkan pengiriman pesan (message) antar objek. Diagram ini dapat digunakan juga untuk menggambarkan suatu *use case* secara detail, dengan kata lain menggambarkan sebuah detail dari suatu proses bisnis yang terjadi di dalam sistem. Pada sebuah *sequence diagram* terdapat obyek yang terdapat pada suatu proses bisnis dan interaksi mereka satu sama lain.



Gambar 2. 7 Contoh *Sequence Diagram*

d. *Class Diagram*

Class diagram untuk memodelkan struktur kelas dan hubungan dari entitas yang terdapat di dalam sebuah sistem. Sebuah kelas diagram digambarkan sebagai berikut:



Gambar 2. 8 Contoh *Class Diagram*

Dimana setiap kotak merepresentasikan kelas dan bagian atas merupakan nama dari kelas tersebut, bagian tengah berisi atribut-atribut yang dimilikinya, sedangkan bagian bawah berisi metode atau operasi yang ada pada kelas tersebut. Anak panah dan garis yang menghubungkan antar kelas menggambarkan relasi antara kelas yang dihubungkan. Angka yang terdapat pada kedua sisi penghubung antar kelas menunjukkan multiplicity dari kedua kelas tersebut.

2.2.7 Microsoft Visual Studio

Visual studio .Net adalah sekumpulan alat pengembangan *software* yang diperuntukkan bagi .Net platform[9]. 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++.

2.2.8 MySQL

MySQL adalah *multiowner 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 [10].