BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tahap tinjauan perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan di PD. Axa.

2.1.1 Sejarah PD. Axa

PD. Axa dahulunya bernama Toko AN. Di dirikan pada tahun 1950 oleh Kinaria Santosa Enjang Putra yang bertempat di jalan Otto iskandar dinata No. 157. Toko AN bergerak di bidang perdagangan yaitu Toko kertas, yang dikelola oleh keluarga.

Setelah pisah toko AN pindah pada tahun 1980 ke jalan Jendral Sudirman Bandung No. 55B Bandung. Selain berjualan kertas, Toko AN juga menjual alat – alat tulis dan selain itu juga memproduksi buku – buku tulis yang di kenal dengan nama cap seriring, selain buku tulis juga mulai memproduksi D/F garis, Stopmap, Kertas stensil, Kertas Durslah dan HVS potongan.

Toko AN buka cabang di jalan Cibadak dengan nama Anjaya dan Cap Seriring. Di rubah menjadi AN pada tahun 1990. Toko AN pecah menjadi 2 (dua) yang satu di cibadak dengan nama Anjaya dan yang satu di jalan Jendral Sudirman Bandung dengan nama Toko AN.

Tahun 1997 nama Toko AN diganti dengan nama AXA, nama AXA diambil dari kata Aksara yang berati Tulisan. Axa Stationery bergerak di bidang alat – alat tulis kantor dan juga kertas, Selain itu juga AXA Stationery mensuplai barang – barang ke pabrik – pabrik dan juga ke kantor – kantor, pemerintahan. Seperti ke PT. INTI (Telekomunikasi), KEWALRAM (Kain Sutera), ATEJA, LEN (Alat – Alat Pemancar), INDOMARCO, SAPTA SARI (Farmasi) dll.

Saat ini PD. AXA Bandung memiliki 3 (tiga) tempat dengan memiliki ruang lingkup masing – masing, yang membedakan ruangan yang satu dengan yang lain dilihat dari perbedaan barang yang akan dijual.

PD. AXA Bandung memiliki 3 (tiga) tempat sebagai perusahaan perdagangan yaitu Bagian Depan menjual berbagai macam kertas minya, kertas kado, dan alat – alat tulis lainnya. Tetapi bagian ini tidak menjual dengan harga eceran atau satuan. Kemudian Bagian Gambar menjual berbagai macam alat – alat tulis gambar seperti pensil warna, buku gambar dan lain – lain. Dan yang terakhir Bagian Atas menjual berbagai alat – alat tulis kantor, tetapi bagian ini dominan menjual berupa kertas Fotocopy, kertas Continous form dan bagian ini memiliki gudang sebagai tempat penyimpanan barang dan sekaligus sebagai persediaan (stok) barang.

PD. AXA Bandung kini dipimpin oleh istri Kinaria Santosa Enjang Putra yaitu ibu Rosiyani dengan 3 (tiga) kepala yang bertanggung jawab pada masing – masing bagian, Bagian Depan dikepalai oleh ibu Nunung, Bagian Gambar dikepalai oleh ibu Yani, dan yang terakhir bagian Atas dikepalai oleh ibu Rosiyanti. Masing – masing bagian ini bertanggung jawab terhadap masing – masing ruangan dan bertanggung jawab pada kelancaran perusahaan dan karyawan – karyawannya.

Lokasi PD. AXA Stationery terletak di jalan Jendral Sudirman No. 55B di samping Bank Buana Bandung. Selain itu memiliki 7 (tujuh) ruangan, diantaranya Ruang Depan, Ruang Kantor, Ruang Gambar, Ruang Atas, Bagian Gudang, Bagian Teknisi, Ruang Delivery. Bentuk bangunan PD. AXA Stationery yaitu berbentuk persegi panjang.

2.1.2 Visi dan Misi

PD. Axa memiliki Visi dan Misi sebagai berikut:

A. Visi

Menjadi perusahaan yang mempunyai reputasi baik dalam industri perdagangan.

B. Misi

Menyediakan produk atau barang yang berkualitas tinggi dengan didukung oleh pelayanan profesional dan memiliki manajemen yang baik. Kebijakan mutu perusahaan mempunyai 5K yaitu : Kerapihan, Ketaatan, Kedisiplinan, Kebersihan, Keterampilan dalam bekerja.

2.1.3 Logo Perusahaan

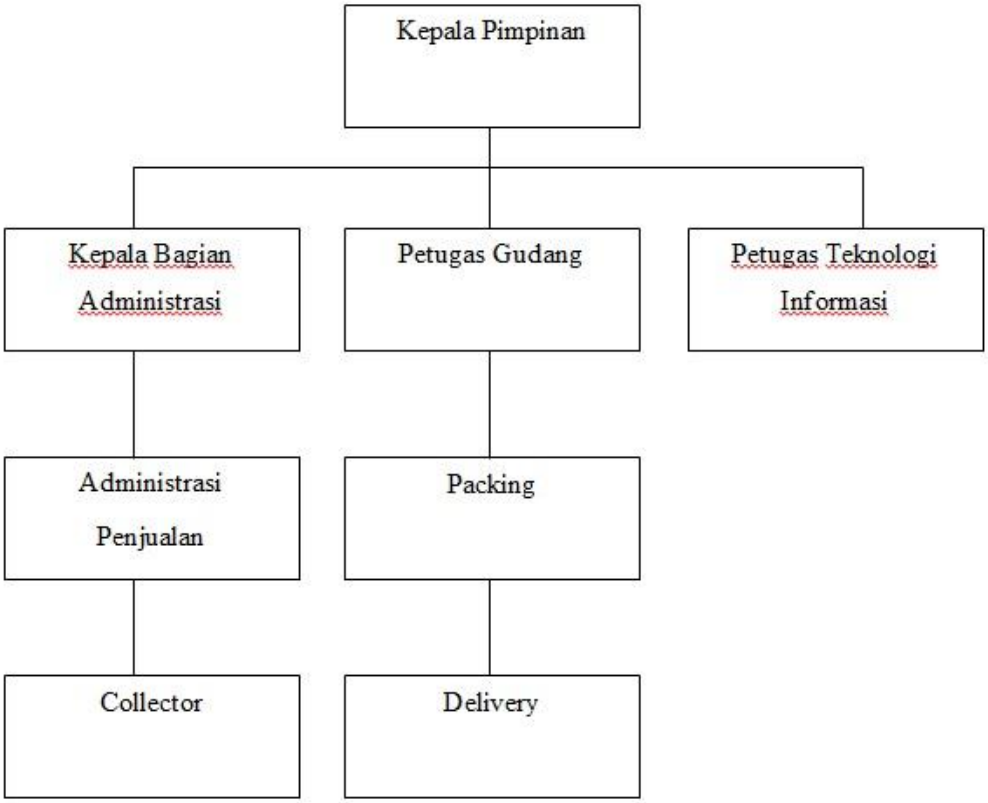
Logo PD. Axa dapat dilihat pada Gambar 2.1



Gambar 2.1 Logo PD. Axa

2.1.4 Struktur Organisasi PD. Axa

Struktur Organisasi yang terdapat pada PD. Axa terdapat pada Gambar 2.2



Gambar 2.2 Struktur Organisasi PD. Axa

2.1.5 Deskripsi Tugas

Berikut adalah deskripsi tugas dari masing-masing jabatan yang tertera dalam struktur organisasi :

1. Kepala Pimpinan

   Bertugas menyusun rencana operasional perusahaan, memeriksa laporan seluruh keuangan perusahaan

2. Kepala Bagian Administrasi

   Melakukan pengadministrasian tentang belanja keluar, kelancaran mekanisme perusahaan secara keseluruhan sesuai urutan tentang kendali dalam struktur organisasi.

3. Administrasi Penjualan

   Membuat surat jalan dalam setiap permintaan *customer* atau pelanggan yang dilampiri Po (*Purchases Order),* mencatat laba dalam setiap penjualan barang dari nota kontan atau dari nota hutang dan menerima pesanan barang.

4. Collector

   Bertugas melakukan penagihan perusahaan (Pembeli) yang telah jatuh tempo.

5. Petugas Gudang

   Melakukan pengecekan, pengawasan barang gudang, memmasukan dan mengeluarkan ke buku gudang.

6. Packing

   Bertugas dalam mengepak atau menyiapkan barang-barang pesanan, mencatat barang-barang yang kurang dan stok barang.

7. Delivery

   Mengirimkan barang yang telah disiapkan oleh packing untuk dikirim ke customer, mencatat buku laporan tempat tujuan yang akan dikirim

8. Petugas Teknologi Informasi (IT)

   Bertugas melakukan segala sesuatu yang berhubungan dengan teknologi informasi.

2.2 Landasan Teori

Dalam penelitian skripsi ini digunakan landasan teori yang berisi teori - teori yang bisa dijadikan dasar dan acuan dalam perancangan sistem CRM di PD. Axa.

2.2.1 Sistem Informasi

Istilah sistem berasal dari bahasa yunani yaitu sistema. Ditinjau dari asal katanya, sistem berarti sekumpulan objek yang bekerja bersama-sama untuk menghasilkan suatu kesatuan metode yang digabungkan dan diatur sedemikian rupa yang berfungsi mencapai tujuan. Suatu sistem dapat berdiri dari beberapa sub-sistem yang saling berhubungan membentuk suatu kesatuan sehingga sasaran atau tujuan sistem tersebut dapat dicapai. Pada umumnya ciri-ciri sistem bertujuan, punya batas, terbuka, tersusun dari subsistem dan saling berkaitan dan saling ketergantungan, merupakan suatu kebulatan yang utuh, melakukan kegiatan transformasi, ada mekanisme kontrol dan memiliki kemampuan mengatur dan menyesuaikan diri sendiri [1].

Informasi merupakan data yang telah diproses, transformasi data, sehingga berubah bentuk menjadi informasi. Informasi adalah data yang telah diambil kembali, diolah atau sebaliknya digunakan untuk tujuan informatif, argumentasi ataupun sebagai dasar untuk pengambilan keputusan. Informasi adalah hasil proses data yang bentuknya kurang berguna menjadi data yang berguna. Kualitas suatu sistem informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan [1].

1. Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.
2. Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Kegunaan informasi itu sendiri adalah untuk memberitahukan kepada pengguna informasi mengenai suatu masalah agar pengguna informasi lebih dapat menguasai masalah yang

dihadapinya. Informasi juga tentang suatu masalah yang dapat digunakan untuk memilih resiko yang paling kecil dan keuntungan yang besar dalam pemilihan alternatif bagi suatu proses pengambilan keputusan.

Dari definisi sistem dan informasi diatas, maka dapat disimpulkan bahwa sistem informasi adalah suatu sistem didalam organisasi yang merupakan kombinasi dari orang, aktivitas, fasilitas, teknologi, media, prosedur dan pengendalian, yang ditujukan untuk mendapatkan jalur komunikasi penting menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdas.

2.2.2 Customer Relationship Management

Definisi CRM dapat dilihat dari berbagai sisi yaitu sisi yang berkaitan dengan teknologi, sisi komunikasi dan manajemen, dan sisi bisnis. Pengertiannya adalah sebagai berikut[2]:

1. CRM adalah “sebuah strategi untuk mengoptimalkan nilai lifetime pelanggan(customer lifetime value) dengan cara mengetahui lebih banyak mengenai informasi pelanggan dan berinteraksi dengan pelanggan secara intensif(dari sisi teknologi).

2. CRM didefinisikan sebagai “sebuah pendekatan perusahaan untuk memahami dan mempengaruhi perilaku pelanggan melalui komunikasi yang intensif dalam rangka meningkatkan akuisisi pelanggan, mempertahankan pelanggan, loyalitas pelanggan(dari sisi komunikasi dan manajemen).

3. Definisi CRM jika dilihat dari segi bisnis dapat diartikan sebagai “sebuah strategi bisnis untuk memahami, mengantisipasi dan mengelola kebutuhan pelanggan yang potensial dalam suatu organisasi pada saat sekarang dan yang akan dating(dari sisi bisnis).

2.2.3 Jenis CRM

Jenis dari *Customer Relationship Management*(CRM) adalah operasional CRM, analitikal CRM dan *collaborative* CRM[3], yaitu :

1. Operasional CRM

   Operational CRM berkaitan dengan fungsi bisnis suatu perusahaan yang meliputi servis pelanggan, manajemen pesanan, invoice atau billing, manajemen dan otomatisasi penjualan.

2. Analitikal CRM

   suatu aktifitas yang meliputi menyimpan, memindahkan, memproses, mengartikan dan melaporkan data Pelanggan atau pengguna, yang kemudian menganalisa apa yang mereka perlukan.

3. *Collaborative* CRM

   Collaborate CRM adalah aktifitas untuk menggabungkan semua komunikasi yang diperlukan, koordinasi dan kolaborasi antara vendors dan customer.

2.2.4 Arsitektur CRM

Arsitektur atau kerangka kerja CRM adalah prespektif pelanggan dan organisasi dalam penerapan CRM, Arsitektur CRM terbagi kedalam 2 jenis yakni [4] :

1. Front CRM

   Front-CRM merupakan kerangka kerja (Framework) untuk membangun sistem komputer untuk mengelola hubungan pelanggan (CRM) menurut perspektif internal organisasi. Tantangan terbesar adalah bagaimana konsep CRM dapat diimplementasikan pada suatu perusahaan. Masalahnya, konsep CRM bukan hanya sekedar instalasi perangkat lunak semata, namun lebih menekankan diterapkannya filosofi dan budaya *customer-centric* secara menyeluruh.

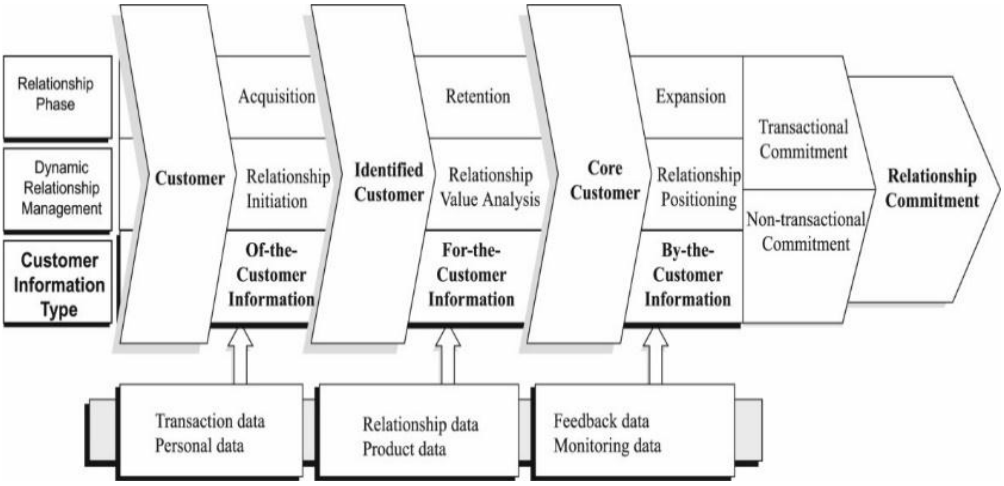
   Penerapan CRM harus dimulai pada jenjang strategi bisnis, di mana pada tingkat ini perubahan organisasi secara menyeluruh dimungkinkan. Agenda perubahan kemudian diarahkan kepada proses kerja, dan didukung oleh teknologi informasi. Proses ini tidak dapat

berjalan sebaliknya. Faktanya, mayoritas proyek implementasi CRM pertama kali difokuskan kepada aspek teknologi daripada strategi bisnis. Hal ini dapat menjadi salah satu faktor penyebab gagalnya implementasi CRM.

Front-CRM dimaksudkan untuk memberikan kerangka kerja untuk dapat mengenali proses bisnis apa saja yang terjadi dalam setiap bidang manajemen yang berhubungan dengan CRM, terutama pada aspek pemasaran, penjualan, dan pelayanan (*departemental plans*). Kerangka kerja disusun berdasarkan aktivitas Perencanaan Strategis, Pemasaran, Penjualan, dan Pelayanan yang terdapat pada perusahaan. Keempat aktivitas inilah merupakan filosofi dasar untuk menerapkan CRM. Untuk dapat lebih memahami apa yang menjadi esensi pada setiap departemen terkait dengan perencanaan strategis dan konsep CRM.

2. Dynamic CRM

Penelitian tentang kerangka kerja CRM yang dilakukan oleh Chung-Hoon Park and Young-Gul Kim, meninjau kaitan strategi informasi dan marketing, dan membuat suatu model yang digambarkan dalan "*Framework of Dynamic CRM*". Digambarkan bahwa fase CRM dalam suatu kerangka kerja berupa rangkaian proses *Acquisition-Retention-Expansion*. Sebagai sebuah kerangka kerja CRM yang terintegrasi yang didasari oleh tipe-tipe informasi mengenai pelanggan yang tersusun secara bertingkat sesuai dengan perkembangan hubungan pelanggan.



Gambar 2.3 Kerangka Dynamic CRM[4]

Pada kerangka kerja diatas digambarkan bahwa terdapat tiga tipe informasi pelanggan (*Customer Information*)[4]. Antara lain :

1. *Of- the – Customer Information* (informasi mengenai pelanggan). Informasi mengenai data personal dan data transaksi pelanggan. Informasi dalam kategori ini sering dikenal sebagai *database marketing* dan target *marketing*.
2. *For - the – Customer Information* (informasi untuk pelanggan). Informasi mengenai produk, layanan, dan informasi perusahaan yang dianggap penting oleh pelanggan. Informasi ini bisa saja disalurkan melalui berbagai media komunikasi dalam bentuk *direct mail, auto response system*, dan situs *internet*.
3. *By - the – Customer Information* (informasi oleh pelanggan). Informasi feedback nontransaksional dari pelanggan yang meliputi komplain, usulan, klaim, kebutuhan pelanggan dan lain-lain. Informasi ini biasanya diwujudkan dalam bentuk data pelanggan yang diperluas karena informasi ini sangat membantudalam interaksi pelanggan. Selain itu, karena mengandung komplain, kebutuhan dan saran, informasi tipe ini dapat diaplikasikan untuk mengembangkan produk.

Secara garis besar, Dynamic-CRM sebenarnya menggambarkan serangkaian fase yang harus dilakukan suatu organisasi dalam mencapai manajemen pelanggan yang sesuai dengan bisnis utama perusahaan tersebut. Kerangka kerja ini bisa dianggap *pattern* atau model acuan suatu organisasi dalam meningkatkan maturitas manajemen pelanggan mereka. Dalam kerangka kerja tersebut terdapat beberapa fase antara lain :

1. *Relationship Initiation*

   Pada tahap inisiasi hubungan dengan pelanggan, organisasi melakukan pendataan terhadap pelanggan, biasanya dengan pendaftaran keanggotaan. Setelah mendata pelanggan, organisasi dapat menyimpulkan informasi tentang penawaran apa yang tepat bagi pelanggan tersebut *(for-the-customer)*. Setelah periode tertentu pelanggan merasa puas dengan apa yang ditawarkan organisasi, maka hubungan dengan pelanggan bisa meningkat menjadi *core customer*. Pada tahap ini organisasi dapat melanjutkan fase berikutnya yaitu CRM. Pada fase ini terjadi interaksi dua arah antara pelanggan dan organisasi. Pelanggan akan mengatakan apa yang dibutuhkan secara spesifik *(by-the-customer)*. Hal seperti ini membuka kesempatan bagi organisasi untuk menawarkan produk atau servis baru pada pelanggan, meningkatkan proses bisnis dan memuaskan *core customer*.
2. *Relationship Value Analysis*

   Agar CRM lebih efektif, sebaiknya setiap hubungan pelanggan dianalisis dari kedua sudut pandang baik pelanggan atau organisasi. Ada dua dimensi dalam menentukan nilai dari setiap hubungan yaitu nilai pelanggan dari sudut pandang dan hak pelanggan dari sudut pandang organisasi. Pada fase ini terjadi hubungan secara psikologis antara pelanggan dengan organisasi. Biasanya ini terjadi dengan pemberian diskon pada pelanggan saat melakukan pembelian dalam jumlah besar, secara tidak langsung loyalitas pelanggan meningkat.

Customer value adalah kumpulan dari benefit relationship yang menguntungkan secara ekonomi dan psikologis.

3. *Relationship Positioning*

   Membangun hubungan yang saling menguntungkan sangat penting untuk menjaga hubungan jangka panjang antara pelanggan dan organisasi. Dalam penerapan strategi membangun hubungan saling menguntungkan, organisasi memaksimalkan hak pelanggan secara otomatis akan meningkatkan nilai pelanggan. Selain itu agar tetap adil organisasi juga harus mengevaluasi hubungan dengan pelanggan dari sudut pandang benefit yang didapatkan. Jika hubungan seperti ini terus berlanjut maka antara customer dan organisasi akan terbentuk komitmen yang kuat baik transactional maupun nontransactional dan hubungan berkembang ke tahap relationship commitment.

2.2.5 Association Rule

"Menurut Jiawei Han and Micheline Kamber, Association rule adalah mencari pola – pola, asosiasi, korelasi, atau struktur kausal antara set item dalam database transaksi, database relasional dan repositori dan lainnya".

"Menurut Kusrini dan Emha Taufiq Luthfi, association rule adalah teknik data mining untuk menemukan aturan asosiatif antara suatu kombinasi item."

Association rule mempunyai parameter, yaitu *support, confidence dan correlation*. Penggunaan *parameter support* dan *confidence* hanya untuk assosiasi data yang menghasilkan beberapa aturan dalam menentukan metode. Namun, kita dapat meningkatkan hasilnya dengan *parameter correlation* [5].

Dalam *association rule*, suatu kelompok *item* dinamakan *itemset*. *Support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *itemset* layak untuk dicari *confidence*-nya (misal, dari keseluruhan transaksi yang ada,

seberapa besar tingkat dominasi suatu *item* yang menunjukkan bahwa *item* A dan item B dibeli bersamaan).*Support* dari *itemset* X adalah persentase transaksi di D yang mengandung X, biasa ditulis dengan *supp(X).* Jika *support* suatu *itemset* lebih besar atau sama dengan *minimum support*, maka *itemset* tersebut dapat dikatakan sebagai *frequent itemset*, yang tidak memenuhi dinamakan *infrequent*. *Confidence* adalah suatu ukuran yang menunjukkan hubungan antara dua*item* (misal, menghitung kemungkinan seberapa sering item B dibeli oleh pelanggan jika pelanggan tersebut membeli sebuah item A). *Confidence* suatu *rule* R (X=>Y) adalah proporsi dari semua transaksi yang mengandung baik X maupun Y dengan yang mengandung X, biasa ditulis sebagai *conf(R).* Sebuah *association rule* dengan *confidence* sama atau lebih besar dari minimum *confidence*Y dapat dikatakan sebagai valid *association rule*.

Aturan asosiasi biasanya dinyatakan dalam bentuk :

{roti, mentega} -> {susu} (support = 40%, confidence = 50%)

Aturan tersebut berarti, 50% dari transaksi di *database* yang memuat *item* roti dan mentega juga memuat *item* susu. Sedangkan 40% dari seluruh transaksi yang ada di database memuat ketiga *item* itu. Dapat juga diartikan, bila seorang konsumen membeli roti dan mentega punya kemungkinan 50% untuk juga membeli susu. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini.

Metode dasar analisis asosiasi tebagi menjadi dua tahap :

1. Analisis pola frekuensi tinggi

   Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Menurut Larose yang dikutif oleh Kusrini, kita bebas menentukan nilai *minimum support* (minsup) dan *minimum confidence* (mincof) sesuai kebutuhan (Kusrini; Luthfi, Emha Taufiq;, 2009). Sebagai contoh, bila ingin menemukan data-data yang memiliki hubungan asosiasi yang kuat, minsup dan

mincofnya bisa diberi nilai yang tinggi. Sebaliknya, bila ingin melihat banyaknya variasi data tanpa terlalu mempedulikan kuat atau tidaknya hubungan asosiasi antara item-nya, nilai minsup dan mincofnya dapat diisi rendah.

Nilai *support* sebuah *item* diperoleh dengan rumus berikut.

$$Support(A) = \frac{Jumlahtransaksimengandung A}{Totaltransaksi}$$

Persamaan 1 menjelaskan bahwa nilai *support* didapat dengan cara membagi jumlah transaksi yang mengandung item A (satu item) dengan
jumlah total seluruh transaksi.Sementara itu, nilai *support* dari 2 item diperoleh dari rumus 2 berikut:

$$Support(A,B) = P(\text{A}\cap\text{B})$$

Support Count adalah banyaknya *itemsets* yang sama muncul secara bersaman pada suatu data transaksi pada keranjang belanja

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan,barulah dicari *association rule* yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif A → B dari support pola frekuensi tinggi A dan B. Nilai *confidence* dari aturan A → B diperoleh dengan rumus berikut.

$$Confidence = P(B|A) = \frac{\sum Transaksimengandung AdanB}{TransaksimengandungA}$$

2.2.6 Parameter Corellation

Colleration merupakan parameter yang menentukan nilai keterhubungan *Parameter correlation* dicari menggunakan rumus *lift.* Nilai *lift* mengukur

seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence.* Terjadinya *itemset* A adalah independen dari terjadinya *itemset* B jika P(A|B)=P(A)P(B), jika tidak, *itemset* A dan B tergantung dan berkorelasi sebagai peristiwa.

Lift yang terjadi antara A dan B dapat diukur dengan menghitung :

$$lift = (A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Dengan aturan sebagai berikut :

1. *Lift* (A,B) < 1 maka membeli (B)
2. *Lift* (A,B) > 1 maka membeli (A,B)

Sebuah transaksi dikatakan valid jika mempunyai nilai *lift* lebih dari 1, yang berarti bahwa dalam transaksi tersebut *item* A dan *item* B benar – benar dibeli secara bersamaan.

2.2.7 Algoritma Apriori

Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian *frequent itemset* untuk mendapatkan *association rules*. Sesuai dengan namanya, algoritma ini menggunakan pengetahuan mengenai *frequent itemset* yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Apriori menggunakan pendekatan secara *iterative* yang disebut juga sebagai *level-wise search* dimana *k-itemset* digunakan untuk mencari (k+1)-*itemset*.

Langkah – langkah algoritma apriori sebagai berikut :

1. Set k=1 (menunjuk pada itemset ke-1).
2. Hitung semua k-*itemset* (itemset yang mempunyai k item).
3. Hitung *support* dari semua calon *itemset. Filter itemset* tersebut berdasarkan nilai minimum support-nya.
4. Gabungkan semua k-sized itemset untuk menghasilkan calon itemset k+1.
5. Set k=k+1 (menunjuk pada itemset ke-2 dan seterusnya)
6. Ulangi langkah 3-5 sampai tidak ada *itemset* yang lebih besar yang dapat dibentuk.

7. Buat final *set* dari *itemset* dengan menciptakan suatu *union* dari semua k-*itemset*.

Untuk meningkatkan efisiensi dari pencarian k-itemset, maka digunakan suatu metode yang dinamakan apriori property, metode ini dapat mengurangi lingkup pencarian sehingga waktu pencarian dapat dipersingkat. Pengertian apriori property adalah suatu kondisi dimana semua subset dari frequent itemset yang tidak kosong haruslah juga frequent, frequent yang dimaksud disini adalah semua itemset yang memenuhi minimum support yang ditentukan oleh pengguna. Berdasarkan definisi yang disebutkan, jika suatu itemsetI tidak memenuhi minimum support (min_sup) yang ditentukan, maka I tidak frequent, atau dinotasikan dengan P(I)<min_sup. Jika suatu *item* A dimasukkan kedalam I, maka hasil *itemset* I U A tidak dapat terjadi lebih dari jumlah itemset I. Oleh karena itu I U A tidak *frequent* juga, dengan kata lain P(I U A)<*min_sup*. Secara sederhana dapat dikatakan jika suatu *itemset* tidak memenuhi syarat, maka semua *superset*(*itemset* lain yang dibentuk dengan mengandung unsur *itemset* tersebut) dari *itemset* tesrebut tidak mungkin akan memenuhi syarat juga.

Berikut penjelasan lebih lanjut mengenai algoritma *apriori property* yang digunakan dalam algoritma ini. Penjelasan berikut merupakan proses yang terdiri dari dua langkah yaitu *join* dan *prune* (Han, Jiawei; Kamber, Micheline, 2006):

1. Langkah join : untuk menentukan L_k, suatu set kandidat k-*itemset* dihasilkan dengan cara menjoinkan L_k-1 dengan dirinya sendiri. Set kandidat ini dihasilkan sebagai Ck. Misalnya I1 dan I2 adalah itemset di dalam L_k-1. Sebagai contoh I1[k-2] mengacu pada kedua untuk item terakhir di I1. Perlu ditekankan bahwa algoritma ini mempunyai aturan bahwa item di dalam transaksi atau itemset telah diurutkan secara lexicographic order terlebih dahulu. Selanjutnya join yang dinotasikan dengan L_k-1 ∞ L_k-1 dijalankan, dimana anggota-anggota L_k-1 yang berupa itemset-itemset dapat di-join jika first (k-2)-items dari L_k−1 sama. Jadi anggota l_1 dan l_2 dari L_k-1 akan di-join jika (l_1[1]=I_1[1])^(I_1[2] = I_1[2]) ^…^ (I_1[k-2] = I_2[k-2]) ∧ (I_1[k-1] < I_2[k-1]). Kondisi (I_1[k-1] < I2[k-1]) bertujuan untuk memastikan tidak ada

duplikat itemset yang dihasilkan dari join I_1 dan I_2 yaitu dalam format $I_1[1]$, $I_1[2]$,…,$I_1[k-2]$, $I_1[k-1]$, $L_2[k- 2]$.

2. Langkah prune : C_k adalah superset dari L_k dimana setiap anggotanya bisa frequent ataupun tidak, tetapi semua frequentk-itemset termasuk dalam C_k. Proses scan terhadap database yang dilakukan untuk menentukan jumlah kemunculan setiap candidate yang ada di dalam C_k akan menentukan L_k (sebagai contoh, semua candidate yang memiliki support count tidak kurang atau sama dengan minimum support yang ditentukan disebut frequent, yang artinya juga memenuhi syarat untuk masuk menjadi L_k). Ada kemungkinan C_k dapat mengandung candidate dengan jumlah yang sangat besar, sehingga dapat menyebabkan proses perhitungan C_k menjadi berat. Untuk mengurangi jumlah candidate C_k, apriori property digunakan sebagai berikut.Semua (k-1) itemset yang tidak frequent 19 dinyatakan tidak dapat menjadi subset dari sebuah frequent k-itemset. Oleh karena itu, jika ada (k-1)-subset dari sebuah candidate k-itemset tidak termasuk dalam L_k-1, maka candidate tersebut tidak mungkin frequent juga dan oleh karena itu dapat di-remove dari C_k.

Berikut ini diberikan penjelasan lebih lanjut melalui contoh kasus, pemakaian algoritma *apriori* untuk menemukan *association rule*. Contoh tabel berikut adalah merupakan transaksi *database* D dari sebuah toko. Ada sembilan transaksi didalam *database* ini.

Tabel II-1 Contoh tabel *Transaction Database* D

Kode Transaksi	Item yang dibeli
T100	I1,I2,I5
T200	I2,I4
T300	I1,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3

T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Berikut adalah langkah-langkah algoritma apriori untuk menemukan *frequent itemset :*

a. Pada iterasi dari algoritma, setiap item adalah anggota set dari *candidate* 1-*itemset*, C_1. Algoritma akan secara langsung memeriksa semua transaksi yang ada di dalam *database* D untuk dapat menghitung kejadian munculnya setiap *item*.
b. Jika diasumsikan bahwa *minimum support* yang dibutuhkan adalah 2 (dalam hal ini *min_sup*=2/9=22%). Set dari *frequent* 1-*itemset* dapat diperoleh dari semua candidate 1-*itemset* atau C_1 yang memenuhi 20 minimum *support*. Dalam contoh ini kebetulan semua anggota C_1 memiliki *support count* ≥ *minimum support*. Jadi tidak ada anggota C_1 yang di hapus pada L_1.
c. Untuk menemukan frequent 2-*itemset* atau L_2, pertama-tama algoritma ini menggunakan join $L_1 \infty L_1$ untuk menghasilkan *candidate* 2-*itemset* atau C_2.
d. Kemudian, transaksi yang ada pada *database* D diperiksa atau di-scan dan *support count* dari tiap *candidate itemset* yang ada di C_2 ditambahkan, seperti yang ditunjukkan pada Gambar II-4. Dapat dilihat pada kolom sebelah kanan,setiap kandidat *itemset* yang ada dalam C_2 tertera support count dari masing-masing candidate 2-*itemset*.
e. Set dari 2-*itemset* atau L_2, ditentukan dari semua candidate 2-*itemset* yang memenuhi minimum support. Dapat dilihat pada Gambar II-4 bahwa itemset pada C_2 yang memiliki support count lebih kecil dari 2 dihapus sehingga yang tersisa adalah *itemset* yang memiliki *support count* ≥ *minimum support*.

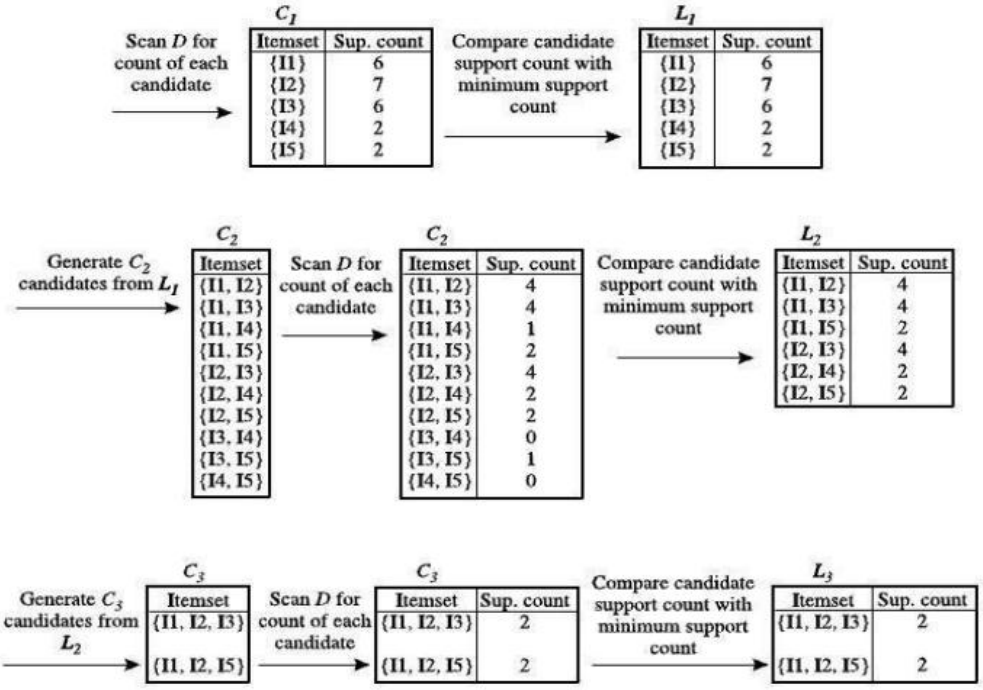
f. Proses untuk menghasilkan atau men-*generate* set dari candidate 3-*itemset* atau C_3, dijelaskan secara lebih *detail* dalam dua langkah yaitu *join* dan *prune* di dalam *apriori property*.

   Join step: Langkah pertama di dalam mendapatkan C_3, yaitu dengan mengkombinasikan L_2 dengan L_2 melalui proses *join* $L_2 \infty L_2$, menghasilkan {{I_1,I_2,I_3},{I_1,I_2,I_5}, {I_1,I_3,I_5}, {I_2,I_3,I_4}, {I_2,I_3,I5}, {I_2,I_4,I_5}}. Detail dari $L_2 \infty L_2$ dapat dilihat pada Gambar II-4.

   Prune step: Berdasarkan pada ketentuan *apriori property* yang mengatakan bahwa semua *subset* dari masing-masing *frequentitemset* diatas juga harus *frequent*, maka dapat ditentukan bahwa keempat *candidate* terakhir yaitu {{I_1,I_3,I_5}, {I_2,I_3,I_4}, {I_2,I_3,I_5}, {I_2,I_4,I_5}} tidak mungkin *frequent*. Oleh karena itu, keempat *candidate* tadi harus di-*remove* dari C_3. Dengan demikian dapat menghemat waktu yang tidak diperlukan untuk melakukan *scan* pada *database*, ketika akan menetukan L_3. Perlu diingat, bahwa ketika diperoleh *candidate* k-*itemset* atau Ck setelah langkah *join*, maka perlu di periksa terlebih dahulu, apakah (k-1)- subset dari masing-masing anggota C_k adalah frequent, sehubungan dengan algoritma apriori menggunakan strategi *level-weis search*. Jadi *apriori property* ini dilakukan secara *level-weis search* pada L_k dimana k dimulai dari level 2 yaitu L_2 dan seterusnya.

g. Semua transaksi di dalam D diperiksa atau di-*scan* untuk menentukan L_3 yaitu terdiri dari candidate 3-*itemset* di dalam C3 yang memenuhi *minimum support* yang sudah ditentukan.

h. Algoritma akan melakukan kombinasi antara L_3 dan L_3 untuk menghasilkan set dari *candidate* 4-*itemset*, C4. Walaupun hasil join adalah {I_1,I_2,I_3}, itemset ini di-prune karena salah satu (k-1)-subset-nya yaitu {I_2,I_3,I_5} tidak *frequent*. Dengan demikian C4=Ø , dan jalannya proses algoritma ini dihentikan karena telah menemukan sebuah frequent itemset.



Gambar 2.4 Pencarian *candidate dan frequent itemset* menggunakan *apriori*

2.2.8 PHP

Php adalah singkatan dari "PHP: Hypertext Preprocessor" yang merupakan bahasa pemrograman yang berbentuk scripting, sistem kerja dari program ini adalah sebagai interpreter bukan sebagai compiler. Pada bahasa pemrograman, yang bisa dikatakan sebagai compiler adalah bahasa yang akan mengubah script-script program ke dalam source code, selanjutnya dari bentuk source code akan diubah menjadi bentuk object code, bentuk dari object code akan menghasilkan file yang lebih kecil dari file mentah sebelumnya. Selanjutnya bentuk object code akan menjadi sebuah program yang siap dijalankan tanpa adanya progam bantu pembuatnya, sehingga hasil dari bahasa pemrograman yang berbentuk compiler akan membentuk sebuah program yang berstatus sebagai program EXE yang dapat langsung dijalankan. Contohnya seperti pascal, C ataupun pemrograman yang berbentuk Visual seperti Delphi maupun Visual Basic[6].

Dengan menggunakan PHP, maka maintenence suatu situs web menjadi lebih mudah. Proses update data dapat dilakukan dengan menggunakan aplikasi

yang dibuat menggunakan script PHP. Kemampuan PHP yang paling diandalkan dan signifikan adalah dukungan kepada banyak database. Membuat halaman web yang menggunakan data dari database dengan sangat mudah dapat dilakukan.

2.2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia.

MySQL adalah Relational *Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*).

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu system database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase[6].

2.2.10 Internet

Internet (*interconnected computer networks*) bisa didefinisikan network komputer tiada batas yang menjadi penghubung pengguna komputer dengan pengguna komputer lainnya serta dapat berhubungan dengan komputer di sebuah wilayah ke wilayah di penjuru dunia, di mana di dalam jaringan tersebut mempunyai berbagai macam informasi serta fasilitas layanan internet browsing atau surfing.

Internet juga merupakan sistem global jaringan komputer yang berhubungan menggunakan standar *Internet Protocol Suite* (TCP/IP) untuk melayani miliaran pengguna di seluruh dunia. Ini adalah jaringan dari jaringan yang terdiri dari jutaan jaringan pribadi, umum, akademik, bisnis, dan jaringan pemerintah dari lokal ke lingkungan global, yang dihubungkan oleh sebuah kode array yang luas dari teknologi jaringan elektronik, nirkabel, dan optik. Internet juga dapat didefinisikan sebagai interkoneksi seluruh dunia komputer dan jaringan komputer yang memfasilitasi sharing atau pertukaran informasi di antara pengguna[7].