

BAB 2

TINJAUAN PUSTAKA

2.1. Profil Instansi

2.1.1. Sejarah Kafe Kopi Cilik

Kafe Kopi Cilik merupakan sebuah perusahaan penjualan makanan terutama kopi yang berada di Jalan Raya Laswi no 607 Ciparay. Kafe ini dibangun oleh seorang wirausahawan bernama Yeremia Natanael atau yang biasa dipanggil dengan sebutan Koko. Kafe yang dibangun pada akhir tahun 2016 inipun berlokasi cukup strategis karena dekat dengan kantor pemerintahan, sekolah, alun-alun dan *waterboom*. Pemilik membangun kafe ini berawal dari kesukaan pemilik pada kopi, selain itu faktor sedikitnya kafe yang ada di daerah Ciparay pada saat itu juga menjadi sebuah alasan dibangunnya kafe ini. Awalnya kafe ini hanya menjual kopi dan makanan ringan, akan tetapi karena permintaan pemilik yang menginginkan kafe ini untuk menjual lebih banyak menu supaya mendapat keuntungan lebih, kafe inipun menambah list makanan pada daftar menu nya.

2.1.2. Logo

Berikut adalah logo dari Kafe Kopi Cilik yang dapat dilihat pada gambar 2.1 dibawah ini :



Gambar 1.1 Logo Kopi Cilik

2.1.3. Visi dan Misi

1. Visi

Visi dari kafe Kopi Cilik ini adalah sebagai berikut :

- a. Mendirikan tempat nongkrong dan tempat istirahat yang nyaman untuk masyarakat sekitar.
- b. Membuat cita rasa makanan yang *up to date* sesuai dengan kriteria anak muda saat ini.

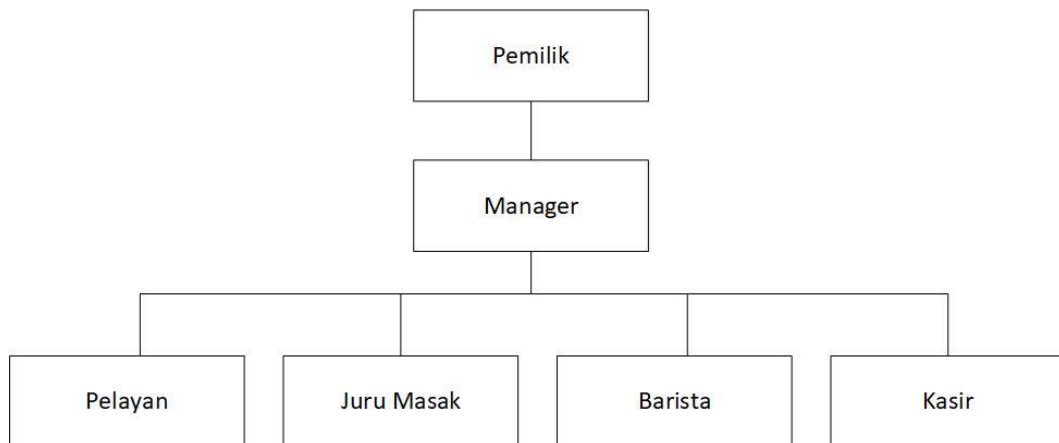
2. Misi

Sedangkan misi dari kafe Kopi Cilik adalah sebagai berikut :

- a. Menyajikan makanan yang berkualitas dengan pelayanan yang ramah.
- b. Selalu berinovasi, baik itu dalam sistem penjualan yang digunakan maupun dalam menu makanan yang ada.
- c. Memenuhi target penjualan sebagai motivasi supaya kafe terus berkembang.

2.1.4. Struktur Organisasi

Struktur organisasi kafe Kopi Cilik dapat dilihat pada gambar 2.2 sebagai berikut :



Gambar 1.2 Struktur Organisasi kafe Kopi Cilik

Deskripsi jabatan :

1. Pemilik memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Bertanggung jawab penuh dalam pengoperasian kafe.
 - b. Memberi arahan serta informasi penting yang berkaitan dengan kafe.

2. Manager memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Merencanakan menu yang akan disediakan.
 - b. Mengatur jadwal kerja karyawan.
 - c. Menyusun anggaran untuk kebutuhan stok dan bahan baku.
 - d. Mempersiapkan acara dan *event* spesial.
 - e. Memantau bahan baku.
3. Pelayan memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Membantu dalam melayani pesanan-pesanan pelanggan.
 - b. Membersihkan meja pelanggan.
 - c. Menyiapkan menu makanan, piring, sendok, gelas, dan lain-lain.
4. Bagian juru masak / koki memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Dapat memasak semua menu baik itu makanan termasuk makanan ringan.
 - b. Bertanggung jawab atas makanan pesanan pelanggan.
 - c. Memanajemen persediaan bahan baku dari menu makanan.
5. Barista memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Dapat membuat semua menu minuman baik kopi maupun minuman lainnya.
 - b. Bertanggung jawab atas minuman pesanan pelanggan.
 - c. Memanajemen persediaan bahan baku minuman atau kopi.
6. Kasir memiliki tugas dan tanggung jawab sebagai berikut :
 - a. Melayani pelanggan dalam pembayaran.
 - b. Bertanggung jawab terhadap hal yang menyangkut transaksi pembayaran.
 - c. Membantu bagian lain jika kafe sedang sibuk atau banyak pengunjung.

2.2. Landasan Teori

Landasan teori yang digunakan berkaitan dengan materi atau teori sebagai acuan untuk melakukan penelitian. Landasan teori yang diuraikan merupakan hasil studi literatur dari buku maupun jurnal.

2.2.1. Data

Secara konseptual, data adalah deskripsi mengenai benda, kejadian aktivitas dan transaksi, yang tidak memiliki makna atau tidak berpengaruh secara langsung kepada pengguna [7]. Data juga bisa disebut sesuatu yang belum memiliki arti sebelum adanya pengolahan dari data itu sendiri. Dalam data terdapat himpunan data yang merupakan kumpulan dari objek beserta atributnya. Atribut merupakan karakteristik dari suatu objek yang biasanya disebut dengan *variable*, *field*, karakteristik atau fitur.

2.2.2. Basis Data

Menurut Fathansyah [8], basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* / tabel / arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Sistem dengan penggunaan basis data mempunyai keuntungan dan kerugian tersendiri, yaitu diantaranya [9] :

Keuntungan :

1. Terpeliharanya keselarasan data
2. Data dapat dipakai secara bersama-sama.
3. Memudahkan penerapan standarisasi dan batas-batas pengamanan.
4. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
5. Program /data *independent*.

Kerugian:

1. Mahal dalam implementasinya.
2. Rumit

3. Penanganan proses *recovery* backup sulit.
4. Kerusakan pada sistem basis data dapat mempengaruhi.

2.2.3. Database Management System

Manajemen sistem basis data adalah perangkat lunak yang di rancang untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar [10]. *Database management system* atau DBMS merupakan perangkat lunak yang digunakan untuk membantu memelihara dan memanfaatkan kumpulan data atau informasi. DBMS juga sering digunakan untuk menyimpan data dalam *file* dan menulis kode aplikasi tertentu untuk penggunaannya. Penggunaan DBMS memiliki beberapa manfaat penting diantaranya adalah [10] :

1. Kemandirian data

Program aplikasi idealnya tidak diekspose pada detail representasi dan penyimpanan data. DBMS menyediakan suatu pandangan abstrak tentang data yang menyembunyikan *detail* tersebut.

2. Akses data efisien

DBMS memanfaatkan berbagai teknik yang canggih untuk menyimpan dan mengambil data secara efisien. Fitur ini sangat penting terutama jika data disimpan pada alat penyimpanan eksternal.

3. Integritas dan keamanan data

Jika data selalu diakses melalui DBMS, maka DBMS dapat memanfaatkan batasan integritas. Misalnya, sebelum menyimpan informasi gaji untuk suatu karyawan DBMS dapat memeriksa bahwa besarnya tidak melebihi anggaran departemen. DBMS juga dapat memanfaatkan kontrol akses yang menentukan data apa yang boleh dilihat oleh kelas pengguna yang berbeda.

4. Administrasi data

Ketika beberapa pengguna berbagi data, keputusan administrasi data dapat memberikan perbaikan yang signifikan. Para ahli yang berpengalaman memahami sifat data yang akan diolah dan memahami

bagaimana kelompok pengguna yang berbeda menggunakan data tersebut, sehingga dapat mengatur representasi data untuk meminimalkan redundansi.

5. Akses konkuren dan *crash recovery*

DBMS menjadwalkan akses konkuren pada data dalam cara tertentu sehingga pengguna data memandang data sebagai data yang sedang diakses oleh hanya satu pengguna pada satu waktu.

6. Waktu pengembangan aplikasi berkurang

DBMS mendukung fungsi penting yang merupakan hal yang bisa bagi banyak aplikasi untuk mengakses data dalam DBMS. Hal ini, dalam kaitannya dengan *interface* level tinggi pada data, memfasilitasi pengembangan aplikasi yang cepat. Aplikasi DBMS kemungkinan besar menjadi lebih akurat dari pada aplikasi yang berdiri sendiri karena banyak tugas penting ditangani oleh DBMS.

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 pengguna. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil.

2.2.4. *Data Mining*

Data mining atau sering disebut juga dengan *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar, keluaran dari *data mining* ini dapat dipakai untuk memperbaiki pengambilan keputusan dimasa depan [11].

Secara umum, definisi *Data Mining* dapat diartikan sebagai berikut [12]:

- a. Proses penemuan pola yang menarik dari data yang tersimpan dalam jumlah besar.

- b. Ekstraksi dari suatu informasi yang berguna atau menarik (*non-trivial*, implisit, sebelumnya belum diketahui potensi kegunaannya) pola atau pengetahuan dari data yang di simpan dalam jumlah besar.
- c. Eksplorasi dari analisa secara otomatis atau semi otomatis terhadap data dalam jumlah besar untuk mencari pola dan aturan yang berarti.

Dari pengertian mengenai *data mining* di atas dapat disimpulkan bahwa *data mining* adalah suatu ilmu yang digunakan untuk mencari suatu pengetahuan dari tumpukan data dengan menggunakan sebuah metode. *Data mining* dapat membantu pengguna dengan menemukan hubungan dan pola antar data dan menyajikannya dengan cara yang dapat dipahami sehingga hubungan tersebut dapat menjadi dasar pengambilan keputusan.

1. Proses Penerapan *Data Mining*

Cross-Industry Standard Process for Data Mining (CRISP-DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industri seperti Daimler Chrysler, SPSS dan NCR telah ditetapkan sebagai proses standar dalam *data mining* yang dapat diaplikasikan di berbagai sektor. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Untuk data yang dapat di proses dengan CRISP-DM ini, tidak ada ketentuan atau karakteristik tertentu, karena data tersebut akan diproses kembali pada fase-fase di dalamnya. Dalam CRISP-DM sebuah pengembangan proyek *data mining* memiliki siklus hidup yang terbagi dalam enam fase [6]. Berikut ini adalah penjelasan mengenai enam fase siklus hidup pengembangan *data mining* [6] :

a. *Business Understanding*

Tahap pertama adalah memahami tujuan dan kebutuhan dari sudut pandang bisnis, dan menerjemahkan pengetahuan ini ke dalam pendefinisian masalah dalam *data mining*. Kemudian menilai situasi pada perusahaan, tahap ini melibatkan lebih rinci fakta tentang sumber daya, asumsi dan faktor-faktor lain yang harus di pertimbangkan dalam menentukan tujuan data dan proyek.

Selanjutnya akan ditentukan rencana dan strategi untuk mencapai tujuan tersebut.

b. *Data Understanding*

Tahap ini dimulai dengan pengumpulan data yang kemudian akan dilanjutkan dengan proses untuk mendapatkan pemahaman yang mendalam tentang data, mengidentifikasi masalah kualitas data, atau untuk mendeteksi adanya bagian yang menarik dari data yang dapat dianalisis menjadi informasi.

c. *Data Preparation*

Tahap ini meliputi semua kegiatan untuk membangun *dataset* akhir (data yang akan diproses pada tahap pemodelan/*modeling*) dari data mentah. Tahap ini dapat diulang beberapa kali. Pada tahap ini juga mencakup pemilihan tabel, record, dan atribut-atribut data termasuk proses pembersihan dan *mapping* data untuk kemudian dijadikan masukan dalam tahap pemodelan (*modeling*).

d. *Modeling*

Dalam tahap ini akan dilakukan pemilihan dan penerapan berbagai teknik pemodelan dan beberapa parameternya akan disesuaikan untuk mendapatkan nilai yang optimal. Secara khusus, ada beberapa teknik berbeda yang dapat diterapkan untuk masalah *data mining* yang sama. Di pihak lain ada teknik pemodelan yang membutuhkan format data khusus. Sehingga pada tahap ini masih memungkinkan kembali ke tahap sebelumnya.

e. *Evaluation*

Pada tahap ini, model sudah terbentuk dan diharapkan memiliki kualitas baik jika dilihat dari sudut pandang analisa data. Pada tahap ini akan dilakukan evaluasi terhadap keefektifan dan kualitas model sebelum digunakan dan menentukan apakah model dapat mencapai tujuan yang ditetapkan pada fase awal (*Business Understanding*). Inti dari tahap ini adalah menentukan apakah ada masalah bisnis yang belum dipertimbangkan. Di akhir dari tahap ini harus ditentukan penggunaan hasil proses *data mining*.

f. *Deployment*

Pada tahap ini, pengetahuan atau informasi yang telah diperoleh akan diatur dan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh pengguna. Tahap *deployment* dapat berupa pembuatan laporan sederhana atau pengimplementasian proses *data mining* yang berulang dalam perusahaan.

2. Metode *Data Mining*

Secara garis besar, Han dalam bukunya menjelaskan bahwa metode *data mining* dapat dilihat dari dua sudut pandang pendekatan yang berbeda, yaitu pendekatan deskriptif dan pendekatan prediktif [13]. Pendekatan deskriptif ialah pendekatan yang mendeskripsikan data inputan. Metode yang termasuk kedalam pendekatan ini adalah:

1. Metode deskripsi konsep/kelas, data dapat diasosiasikan dengan kelas atau konsep. Ada tiga macam pendeskripsian yaitu:
 - a. karakteristik data, dengan membuat *summary* karakter umum atau fitur data suatu kelas target.
 - b. Diskriminasi data, dengan membandingkan *class* target dengan satu atau sekelompok kelas pembandingan,
 - c. Menggabungkan antara karakterisasi dan diskriminasi.
2. Metode *association rule*, yaitu menemukan aturan asosiatif atau pola kombinasi dari suatu *item* yang sering terjadi dalam sebuah data.

Pendekatan kedua adalah pendekatan prediktif, yaitu pendekatan yang digunakan untuk memprediksi, hasil pendekatan ini berupa kelas atau *cluster*. Metode yang termasuk dalam pendekatan ini adalah:

1. Metode klasifikasi dan prediksi, yaitu metode analisis data yang digunakan untuk membentuk model yang mendeskripsikan kelas data yang penting, atau model yang memprediksikan *trend* data. Klasifikasi digunakan untuk memprediksi kelas data yang bersifat kategorial, sedangkan prediksi untuk memodelkan fungsi yang mempunyai nilai kontinu.
2. Metode *clustering*, mengelompokkan data untuk membentuk kelas-kelas baru atau sering disebut *cluster*. Metode *clustering* bertujuan untuk

memaksimalkan persamaan dalam satu *cluster* dan meminimalkan perbedaan antar *cluster*.

3. Tantangan *Data Mining*

Menurut Fajar Astuti Hermawati, ada beberapa tantangan dalam *data mining*, yaitu [1]:

1. *Scallability*, besarnya ukuran basis data yang digunakan.
2. *Dimensionality*, banyaknya jumlah atribut dalam data yang akan diproses
3. *Complex and Heterogeneous data*, data yang kompleks dan mempunyai variasi yang beragam.
4. *Data quality*, kualitas data yg akan diproses seperti data yg bersih dari *noise* dan *missing value*.
5. *Data ownership and distribution*, siapa yang memiliki data dan bagaimana distribusinya.
6. *Privacy preservation*, menjaga kerahasiaan data yg banyak diterapkan pada data nasabah perbankan.
7. *streaming data*, aliran data itu sendiri.

2.2.5. *Association Rules*

Association rules adalah proses pendeteksian kumpulan atribut-atribut yang muncul bersamaan (*co-occur*) dalam frekuensi yang sering dan membentuk sejumlah kaidah dari kumpulan tersebut [1]. *Association rule* akan menemukan pola tertentu pada data untuk mengasosiasikan antar data satu sama lain. Untuk mencari *association rule* dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari *frequent itemset* terlebih dahulu. *Frequent itemset* merupakan sekumpulan *item* yang sering muncul secara bersamaan. Setelah semua pola *frequent itemset* ditemukan, barulah mencari aturan asosiatif atau aturan keterkaitan yang memenuhi syarat yang telah ditentukan.

Bentuk umum dari *association rule* adalah *Antecedent* \rightarrow *Consequent*. Bila kita ambil contoh dalam sebuah transaksi pembelian barang di sebuah minimarket didapat bentuk *association rule* kopi \rightarrow gula. Artinya, bahwa pelanggan yang membeli kopi ada kemungkinan pelanggan tersebut juga akan membeli gula,

dimana tidak ada batasan dalam jumlah *item-item* pada bagian *antecedent* ataupun *consequent* dalam sebuah *rule*.

Dalam menentukan suatu *association rule*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapat dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu:

1. *Support* : suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan data. Misalnya dari keseluruhan data penjualan yang ada, seberapa besar tingkat dominasi suatu *item* yang menunjukkan bahwa *item A* dan *item B* terjual bersamaan.

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

$$\text{Support}(A) = \frac{\text{jumlah transaksi mengandung } A}{\text{Jumlah total transaksi}} \times 100\% \quad \text{Persamaan (2-1)}$$

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

$$\text{Support}(A,B) = (A \cap B) \quad \text{Persamaan (2-2)}$$

$$\text{Support}(A,B) = \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } B}{\text{Jumlah total transaksi}} \quad \text{Persamaan (2-3)}$$

2. *Confidence*: suatu ukuran yang menunjukkan hubungan antara dua *item* atau lebih secara *conditional*. misalnya, menghitung kemungkinan seberapa sering *item B* dibeli oleh pelanggan jika pelanggan tersebut membeli sebuah *item A*.

$$\text{Confidence}(B \rightarrow A) = \frac{\text{Jumlah transaksi mengandung } A \text{ dan } B}{\text{Jumlah transaksi mengandung } A} \quad \text{Persamaan (2-4)}$$

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan nilai minimum kedua parameter tersebut yang ditentukan oleh pengguna. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan sebelumnya, maka pola tersebut dapat disebut sebagai *interesting rule* atau *strong rule*. Sebagai contoh konsumen yang akan membeli kopi dan gula dengan *support* 2% dan *confidence* 60%, *support* 2% menunjukkan bahwa

keseluruhan dari total transaksi konsumen membeli kopi dan gula secara bersamaan yaitu sebanyak 2%. Sedangkan *confidence* 60% yaitu menunjukkan bila konsumen membeli kopi dan pasti membeli gula sebesar 60%.

2.2.6. Algoritma CT-Pro

Algoritma CT-Pro merupakan salah satu algoritma pengembangan dari FP-Growth. Perbedaannya terdapat pada langkah pembuatan *tree* dimana FP-Growth membuat FP-Tree sedangkan CT-Pro membuat *Compressed* FP-Tree (CFP-Tree) [4]. Pada CFP-Tree ini informasi dari sebuah FP-Tree diringkas dengan struktur yang lebih kecil atau ringan, sehingga baik pembentukan *tree* maupun *frequent itemset mining* yang dilakukan menjadi lebih cepat [4]. Pada tahap *Mining* algoritma CT-Pro menggunakan pendekatan *bottom-up* dimana *item* pada *header table* dan CFP-Tree dilakukan *scan* dari jumlah terkecil hingga terbesar [4].

Berikut ini merupakan langkah-langkah algoritma CT-Pro [4] :

1. Menemukan *item* yang *frequent*
 - a. Kumpulkan dan seleksi data yang diperlukan.
 - b. Data-data yang sudah diseleksi kemudian ditransformasi.
 - c. Kemudian masing-masing data diseleksi berdasarkan minimum *support* yang telah ditentukan, kemudian didapat *Item Frequent Table*.
 - d. Masing-masing *item* dihitung frekuensi kemunculannya sehingga dihasilkan *global item table*.
 - e. Data kemudian *dimapping* berdasarkan *index* pada *global item table*.

2. Membuat CFP-Tree

Setelah menemukan *item* yang *frequent* (*item* yang sering muncul) kemudian dilakukan pembangunan CFP-Tree. *Frequent item* yang ada diurutkan sesuai *global item* dari nilai yang terbesar ke terkecil. CFP-Tree adalah *tree* dengan properti sebagai berikut [4]:

- a. CFP-Tree terdiri dari *tree* yang memiliki *root* yang mewakili *index* dari *item* dengan tingkat kemunculan tertinggi dan kumpulan *subtree* sebagai anak dari *root*.

- b. Jika $I = \{i_1, i_2, \dots, i_k\}$ adalah kumpulan dari *frequent item* dalam data penjualan atau data transaksi, *item* dalam data tersebut akan dimasukkan kedalam CFP-Tree dimulai dari *root subtree* yang merupakan i_1 dalam *header table*.
 - c. *Root* dari CFP-Tree merupakan level-0 dari *tree*.
 - d. Setiap *node* dalam CFP-Tree memiliki empat *field* utama yakni *item-id*, *parent-id*, *count* yang merupakan jumlah *item* pada *node* tersebut, dan level yang menunjukkan struktur data *tree* pada *node* tersebut dimulai dari *item* yang terdapat pada *header table* dengan level yang terdapat pada CFP-Tree.
3. Melakukan *Mining*

Setelah tahap pembangunan CFP-Tree dari sekumpulan data transaksi, lakukan pencarian *node* yang berkaitan dengan *item* dimana pencarian dimulai dari *global item* dengan *support count* terkecil sampai *global item* dengan *support count* terbesar karena CT-Pro bekerja *Bottom-Up* pada *Global CFP-Tree*. Dari semua *node* yang ditemukan untuk setiap *item* inilah yang disebut dengan *Local Frequent item* dan digunakan untuk membuat *Local item table* yang pembuatannya dilakukan berdasarkan jumlah *minimum support* yang telah ditentukan. Selanjutnya dibuat *Local CFP-Tree* berdasarkan *Local Item Table* yang terbentuk, kita dapat membuat *frequent pattern*-nya. Dari hasil *frequent pattern* ini akan dilakukan *mining* berdasarkan nilai *confidence*.

2.2.7. *Unified Modeling Language*

UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan dalam memodelkan *software* dan pengembangan sistem [14]. Berikut adalah beberapa model yang digunakan pada penelitian ini untuk menggambarkan sistem dalam UML diantaranya:

1. Diagram *Use Case*
2. Diagram Kelas
3. Diagram Aktivitas

4. Diagram *Sequence*

2.2.7.1. Diagram *Use Case*

Diagram *use case* adalah model fungsional sebuah sistem yang menggunakan aktor dan *use case*. *Use case* adalah layanan (*services*) atau fungsi–fungsi yang disediakan oleh sistem untuk penggunaanya.

Deskripsi Diagram *Use Case*:

1. Sebuah *use case* adalah dimana sistem digunakan untuk memenuhi satu atau lebih kebutuhan pemakai.
2. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, *design testing*, dan dokumentasi.
3. *Use case* menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.
4. *Use case* menentukan nilai yang diberikan sistem kepada pemakainya.

Use case hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem. *Use case* tidak untuk menentukan kebutuhan nonfungsional, misal: sasaran kerja, bahasa pemrograman.

2.2.7.2. Diagram Kelas

Diagram kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Tahapan dari diagram kelas adalah sebagai berikut:

1. Mengidentifikasi objek dan mendapatkan kelas-kelasnya.
2. Mengidentifikasi atribut kelas-kelas.
3. Mulai mengkonstruksikan kamus data.
4. Mengidentifikasi operasi pada kelas-kelas.

Mengidentifikasi hubungan antar kelas dengan menggunakan asosiasi, agregasi, dan inheritance (pewarisan).

2.2.7.3. Diagram Aktivitas

Diagram aktivitas adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

2.2.7.4. Diagram Sequence

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak.

2.2.8. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user, serta menggunakan perintah standar sql [15]. SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan DBASE atau Clipper yang masih menggunakan perintah-perintah pemrograman.

