

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Profil Perusahaan**

Profil KPRI MASSA meliputi meliputi sejarah, visi dan misi, struktur organisasi, dan deskripsi Kerja.

##### **2.1.1. Sejarah Perusahaan**

KPRI MASSA didirikan pada Tanggal 29 Desember 1995 yang berlokasi di Jagaraga Arjasari Leuwisari. KPRI MASSA Adalah organisasi yang bergerak di bidang penyimpanan dan peminjaman dana. Keanggotaan KPRI sendiri yaitu guru – guru yang sudah menjadi PNS (Pegawai Negeri Sipil) pada lingkungan pendidikan Kabupaten Tasikmalaya dan pensiunan.

Nama koperasi diambil berdasarkan rapat pengurus koperasi yaitu KPRI MASSA. Melalui nama itu pun diharapkan koperasi ini bisa maju terus dalam mengembangkan sistem simpan pinjam yang telah ada, serta diharapkan dapat maju terus dalam meningkatkan pelayanan bagi Anggota.

##### **2.1.2. Visi dan Misi Perusahaan**

###### **Visi**

Untuk Menyejahterakan anggota KPRI MASSA

###### **Misi**

- a) Menciptakan kesejahteraan bagi para anggota yang berkesinambungan.
- b) Mengelola Koperasi dan unit usaha secara profesional dengan menerapkan ‘*Good Corporate Governance*’.

##### **2.1.3 Logo Perusahaan**

Logo KPRI MASSA dilihat pada Gambar 2.1

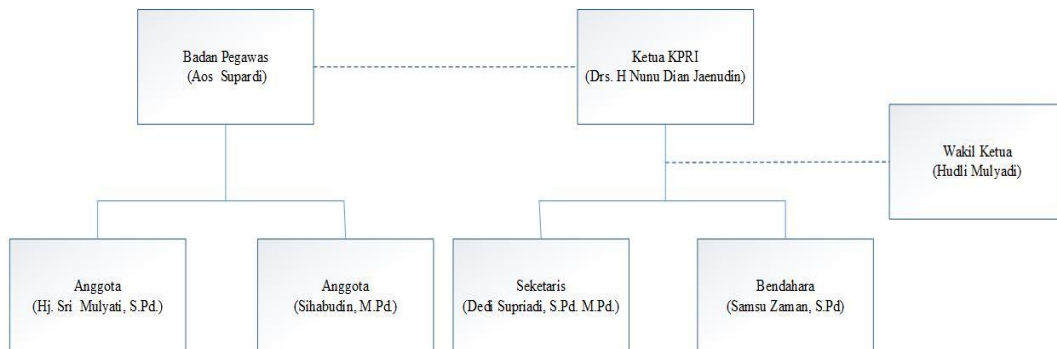


**Gambar 2. 1 KPRI MASSA**

Lambang Koperasi Indonesia memiliki arti sebagai berikut:

1. Rantai melambangkan persahabatan yang kokoh.
2. Gigi Roda melambangkan usaha/karya yang terus menerus.
3. Kapas dan Padi melambangkan kemakmuran rakyat yang diusahakan oleh Koperasi.
4. Timbangan melambangkan keadilan sosial sebagai salah satu dasar koperasi.
5. Bintang dalam perisai melambangkan Pancasila sebagai landasan ideal koperasi.
6. Pohon beringin melambangkan sifat kemasyarakatan dan kepribadian Indonesia yang kokoh berakar.
7. Tulisan Koperasi Indonesia melambangkan kepribadian koperasi rakyat Indonesia.
8. Warna merah dan putih melambangkan sifat nasional Indonesia.
9. Perorangan, yaitu orang yang secara sukarela menjadi anggota koperasi.

### 2.1.4. Struktur Organisasi



**Gambar 2. 2 Struktur Organisasi KPRI MASSA**

Deskripsi Kerja:

#### 1. Ketua KPRI

Ketua KPRI MASSA memiliki tanggung jawab baik ke dalam maupun keluar organisasi, dengan uraian tugas selengkapnya sebagai berikut:

1. Memimpin Koperasi dan mengkoordinasikan kegiatan seluruh anggota Pengurus.
2. Melaksanakan segala perbuatan sesuai dengan Keputusan Rapat anggota dan Rapat Pengurus.

Adapun wewenang dari ketua adalah sebagai berikut :

1. Menentukan Kebijakan dan mengambil keputusan.
2. Menandatangani surat-surat dan perjanjian bersama Sekretaris dan Bendahara.
3. Ketua bertanggung jawab kepada Rapat Anggota.

#### 2. Wakil Ketua

Wakil ketua memiliki wewenang untuk bertindak sebagai wakil penanggung jawab umum, dengan rincian tugas sebagai berikut :

1. Melaksanakan tugas ketua apabila berhalangan.
2. Membina dan mengawasi bidang organisasi dan administrasi.
3. Melaksanakan pendidikan dan penyuluhan.

### 3. Sekretaris

Tugas utama sekretaris adalah sebagai penanggung jawab administrasi koperasi, adapun uraian tugasnya sebagai berikut :

1. Bertanggung jawab kegiatan administrasi.
2. Mengusahakan kelengkapan organisasi.
3. Memimpin dan mengarahkan tugas Anggota.
4. Menghimpun dan menyusun laporan kegiatan bersama bendahara dan pengawas.

Sekretaris berwenang :

1. Mengambil keputusan dibidang kesekretariatan.
2. Menandatangani surat-surat bersama ketua.
3. Sekretaris bertanggung jawab kepada rapat Pengurus melalui Wakil Ketua.

### 4. Bendahara

Pada dasarnya tugas pokok bendahara adalah mengurus kekayaan dan keuangan koperasi, antara lain :

1. Bertanggung jawab masalah keuangan koperasi.
2. Mengatur jalannya pembukuan keuangan.
3. Menyusun anggaran setiap bulan.
4. Mengawasi penerimaan dan pengeluaran uang.
5. Menyusun rencana anggaran dan pendapatan koperasi.
6. Menyusun laporan keuangan.
7. Mengendalikan anggaran.

Bendahara berwenang :

1. Mengambil keputusan dibidang pengelolaan keuangan dan usaha.
2. Bersama dengan ketua menandatangani surat yang berhubungan dengan bidang keuangan dan usaha.

### 5. Badan Pengawas

Di samping rapat anggota dan pengurus, salah satu alat perlengkapan organisasi koperasi adalah pengawas yang antara lain mempunyai tugas untuk melakukan pengawasan terhadap pelaksanaan kebijakan dan pengelolaan koperasi.

Adanya fungsi pengawasan dalam suatu organisasi koperasi, dimaksudkan sebagai salah satu upaya untuk memperkecil risiko yang mungkin timbul sebagai akibat dari terjadinya penyimpangan-penyimpangan kebijakan dari rencana yang telah ditetapkan.

Pengawas dipilih melalui rapat anggota bersama dengan pemilihan pengurus dengan masa jabatan tiga tahun. Jabatan pengawas tidak boleh dirangkap dengan jabatan pengurus, sedangkan persyaratan badan pengawas sama dengan persyaratan pengurus.

Dengan uraian tugas masing-masing adalah sebagai berikut:

1. Melakukan pengawasan terhadap pelaksanaan kebijaksanaan pengurus menyangkut pengelolaan koperasi, baik yang menyangkut aspek organisasi idiil maupun aspek usaha.
2. Meneliti catatan yang ada pada koperasi.
3. Membuat laporan tertulis tentang hasil pengawasan.

## **2.2.Landasan Teori**

### **2.2.1. Data**

Data dapat didefinisikan sebagai bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang tidak acak yang menunjukkan jumlah, tidakkan dapat diolah menjadi bentuk yang lebih kompleks atau hal. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan sebagai file dalam basis data, informasi, database, atau solusi untuk masalah tertentu. Data kan menjadi dalam suatu proses pengolahan data. Oleh karenanya, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut [5].

### **2.2.2. Basis Data**

Basis data (Database) adalah sekumpulan informasi bermanfaat yang diorganisasikan ke dalam tata cara yang khusus.

Database adalah kumpulan 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 [6]:

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 mencerminkan 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 entitinya 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 pengabsisan data.

Penyusunan basis data meliputi proses memasukkan data ke dalam media penyimpanan data, dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (*Database Management System / DBMS*).

### 2.2.3. Data Mining

Proses yang memperkerjakan suatu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstrusi pengetahuan (*Knowledge*) secara otomatis [7]. Definisi lain diantara-Nya adalah pembelajaran adalah berbasis induksi (*induction-based learning*) adalah proses pembentukan definisi-definisi umum yang di lakukan secara mengobservasi contoh-contoh spesifikasi dari konsep-konsep yang akan dipelajari. *Knowledge Discovery Database (KDD)* adalah penerapan metode saintifik pada data *mining* . Dalam konteks ini data *mining* merupakan suatu langkah dari proses KDD [8]. Di bawah ini terdapat beberapa proses dari data *mining* , antara lain:

1. *Deskripsi*

*Deskripsi* bertujuan untuk mengidentifikasi pola yang muncul secara berulang pada suatu data dan mengubah pola tersebut menjadi aturan dan kriteria yang dapat mudah dimengerti oleh para ahli pada domain aplikasinya.

2. *Prediksi*

*Prediksi* memiliki kemiripan dengan klasifikasi, akan tetapi data diklasifikasikan berdasarkan perilaku atau nilai yang diperkirakan pada masa yang akan datang.

3. *Estimasi*

*Estimasi* hampir sama dengan prediksi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi.

#### 4. *Klasifikasi*

*Klasifikasi* adalah proses menemukan sebuah model atau fungsi yang mendeskripsikan dan membedakan data ke dalam kelas-kelas. Klasifikasi melibatkan proses pemeriksaan karakteristik dari objek dan memasukkan objek ke dalam salah satu kelas yang sudah didefinisikan sebelumnya.

#### 5. *Clustering*

*Clustering* adalah pengelompokan data tanpa berdasarkan kelas data tertentu ke dalam kelas objek yang sama. Sebuah kluster adalah kumpulan record yang memiliki kemiripan suatu dengan yang lainnya dan memiliki ketidakmiripan dengan record dalam kluster lain.

#### 6. *Asosiasi*

Tugas asosiasi dalam data *mining* adalah menemukan atribut yang muncul dalam suatu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja (*market basket analysis*). Tugas asosiasi berusaha untuk mengungkap aturan untuk mengukur hubungan antara dua atau lebih atribut.

### **2.2.3. Metode *Cross-Industry Standard Process for Data Mining (CRISP-DM)***

CRISP-DM merupakan siklus hidup proyek data *mining* yang terbagi dalam enam fase seperti di bawah ini. Pengerjaan fase bersifat adaptif, dimana fase berikutnya bergantung pada keluaran fase sebelumnya. Fase yang sudah dikerjakan mungkin dapat kembali pada fase sebelumnya apabila diperlukan evaluasi atau kesalahan pada fase sebelumnya [4].





**Gambar 2. 3 Proses Data Mining menurut CRISP-DM**

Berikut ini adalah enam tahap siklus hidup pengembangan data *mining* (Chapman dkk, 2000) :

1. Fase pemahaman bisnis (*Business Understanding*)  
memahami tujuan dan kebutuhan dari sudut pandang bisnis, kemudian menterjemahkan pengetahuan ini ke dalam pendefinisian masalah pada data *mining* . Selanjutnya akan ditentukan rencana dan strategi untuk mencapai tujuan tersebut.
2. Fase Pemahaman Data (*Data Understanding Phase*)  
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 digunakan untuk heptosa untuk informasi yang tersembunyi.
3. Fase Pengolahan Data (*Data Preparation Phase*)  
meliputi semua kegiatan untuk membangun data set akhir (data yang akan diproses pada tahap permodelan) 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 transformasi data untuk kemudian dijadikan masukan dalam tahap permodelan.

#### 4. Fase Permodelan (*Modeling Phase*)

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 permodelan yang membutuhkan format data khusus. Sehingga pada tahap ini masih memungkinkan kembali ke tahap sebelumnya.

#### 5. Fase Evaluasi (*Evaluation Phase*)

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 mendapat tujuan yang ditetapkan pada fase awal (*Business Understanding*). Kunci dari tahap ini adalah menentukan apakah ada masalah bisnis yang belum dipertimbangkan. Di akhir dari tahap ini harus ditentukan penggunaan hasil proses data *mining*.

#### 6. Fase Penyebaran (*Deployment Phase*)

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 mengimplementasikan proses data *Mining* yang berulang dalam perusahaan. Pada banyak kasus, tahap deployment melibatkan konsumen, di samping analisis data, karena sangat penting bagi konsumen untuk memahami tindakan apa yang harus dilakukan untuk menggunakan model yang telah dibuat.

### 2.2.5. Data Preprocessing

suatu jenis proses yang dapat mengubah data menjadi sebuah informasi atau sebuah pengetahuan. Data Processing ini sering menggunakan komputer sehingga bisa berjalan secara otomatis. Setelah diolah, data ini biasanya mempunyai nilai yang informatif jika dinyatakan dan dikemas secara terorganisir dan rapi, maka

istilah data Processing sering dikatakan sebagai sistem informasi. Langkah dalam data preprocessing.

### 2.2.6. Tahapan Asosiasi

Analisis asosiasi dikenal juga sebagai salah satu teknik Data *mining* yang menjadi dasar dari berbagai teknik Data *mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (*Frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan *Algoritma* yang efisien. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, support (nilai penunjang) yaitu persentase kombinasi item tersebut dalam database dan confidence (Fadlina, 2014) [3].

### 2.2.7. Association Rule

*Association rule* adalah salah satu teknik utama atau prosedur dalam market basket analisis untuk mencari hubungan antar-item dalam suatu data set dan menampilkannya dalam bentuk associaton rule. Dalam pencarian *Association rule* dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari *Frequent item set* terlebih dahulu. *Frequent item set* adalah sekumpulan item yang sering muncul secara bersamaan. Metodologi dasar analisis asosiasi terbagi menjadi 3 (tiga) tahap, yaitu[9]:

1 Analisa 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 [10]:

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

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

$$\text{Support (A U B)} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}}$$

## 2. Pembentukan aturan asosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan asosiatif A\_B Nilai confidence dari aturan A\_B diperoleh dari rumus berikut[10]:

$$\text{Confidence} = P(B | A) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi yang mengandung A}}$$

### 2.2.8. *Frequent Petters*

*Frequent Petters* Adalah pola yang sering terjadi di dalam data. Ada banyak jenis dari *Frequent Petters*, termasuk di dalamnya pola, sekelompok *item set*, *sub-sequence*, dan sub-struktur. Sebuah *Frequent Petters* biasanya mengacu pada satu set item yang sering muncul bersama-sama dalam suatu kumpulan data transnasional, misalnya seperti roti dan susu.

### 2.2.9. **Algoritma Frequent Pattern Growth (FP-Growth)**

*Algoritma Frequent Pattern-Growth* atau yang biasa disebut dengan FP-Growth merupakan pengembangan dari *Algoritma Apriori*, sehingga dalam *Algoritma FP-Growth* ini, segala kekurangan dalam algoritma Apriori telah diperbaiki [11].

(Ririanti, 2014). FP-Growth adalah salah satu alternatif *Algoritma* yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*Frequent item set*) dalam sekumpulan data. *Algoritma FP-Growth* merupakan pengembangan dari *Algoritma Apriori*.

FP-Growth adalah metode yang sering *item set* pertambangan tanpa Generasi calon. Ini membangun sebuah struktur data yang sangat padat (*FP-tree*) untuk kompres database transaksi asli [3].

#### 2.2.9.1. **Langkah-Langkah Proses Perhitungan Association Rule Dengan Algoritma FP-Growth**

Proses perhitungan association rule terdiri dari beberapa tahap adalah sebagai berikut:

##### 1. Membuat Header Item

Heder 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. Heder 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.

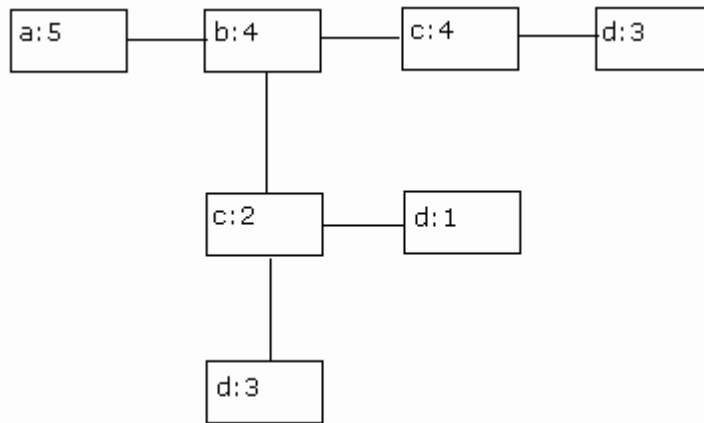
## 3. Pattern Extraction

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 Pattern Tree

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 2.3). 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. 4 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.

### 2.2.10. *Object Oriented Programming (OOP)*

Suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkiriman pesan kepada objek yang lain [12].

Ada pula konsep dari *object oriented programming* adalah sebagai berikut :

#### 1. *Class*

Dalam object oriented program Mining, class (kelas) adalah blueprint atau prototipe, yang mendefinisikan variabel dan method-method pada seluruh object tertentu untuk membuat anggota dirinya sendiri mengarah kepada class instance, class object, instance object atau simpel object

#### 2. *Objek*

*Object* adalah elemen dasar dari konsep pemrograman, merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun object lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya.

#### 3. *Abstraksi*

Abstraksi adalah mengabstrakkan atau penyembunyian kerumitan dari suatu proses.

#### 4. Pembungkusan (*Encapsulation*)

*abstraksi* yang telah kita bicarakan tadi diatas dapat dilakukan dengan cara membungkus semua kode dan data yang berkaitan ke dalam

satu entitas tunggal yang disebut dengan objek, dengan kata lain sebenarnya proses pembungkusan itu sendiri merupakan cara atau mekanisme untuk melakukan abstraksi.

#### 5. Pewarisan (*Inheritance*)

*Inheritance* adalah proses pewarisan data dan method dari suatu kelas kepada kelas yang lain. Pewarisan ini bersifat menyeluruh, sehingga semua data dan method yang dimiliki oleh kelas asalnya akan diturunkan kepada kelas baru. Kelas yang mewarisi disebut kelas “SUPER (super class)” dan kelas yang diwarisi disebut “SUBKELAS (sub class)

#### 6. *Polimorfism*

*Polimorfism* merupakan kemampuan suatu objek untuk mengungkap banyak hal melalui satu cara yang sama.

### 2.2.11. *Unified Modelling Language (UML)*

*Unified Modeling Language (UML)* adalah himpunan struktur dan teknik untuk permodelan desain program berorientasi objek serta aplikasinya. Berikut adalah beberapa model yang digunakan dalam perancangan *Data mining* menentukan besar pinjaman untuk menggambarkan sistem dalam UML [13]:

1. Diagram Use Case
2. Diagram Kelas
3. Diagram Aktivitas
4. Diagram Objek

#### 2.2.11.1. **Diagram Use Case**

Use Case atau diagram use case merupakan permodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakannya fungsi-fungsi itu.

Syarat penamaan pada use case adalah nama definisikan sesimpel mungkin dan dapat dipahami. Ada dua utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case [14].



1. Aktor merupakan orang merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambaran orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor

#### **2.2.11.2. Diagram kelas**

Diagram kelas atau class diagram menggambarkan struktur dari segi pendefinisian kelas-kelas yang akan dibuat untuk mengembangkan sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas dibuat agar pembuatan program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat dengan perangkat lunak sehingga tidak ada gunanya lagi sebuah perancangan apa yang di rancang dan hasilnya tidak sesuai[13].

#### **2.2.11.3. Diagram Aktivitas**

Diagram aktivitas atau activity diagram menggambarkan workflow(aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampil dari sistem/user interface dimana setiap dianggap memiliki sebuah rancangan antar muka tampilan.

3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancang menu yang di tampilkan pada perangkat lunak.

#### **2.2.11.4. Diagram objek**

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalan objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus objek, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan. Diagram objek juga berfungsi untuk mendefinisikan contoh nilai atau isi dari atribut tiap kelas.

Untuk apa mendefinisikan sebuah kelas sedangkan pada ajaran sistem, objek tidak pernah dipakai. Hubungan Link pada diagram objek merupakan hubungan memakai dan dipakai dimana dua buah objek akan dihubungkan oleh yang dipakai oleh objek lain[13].

#### **2.2.12. Java**

Java adalah bahasa pemrograman yang multi platform dan multi device. Sekali ada menuliskan sebuah program dengan menggunakan Java, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis Java ini dikompilasikan ke dalam p-code dan bisa dijalankan dengan Java Virtual Machine. Fungsionalitas dari Java ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik

#### **2.2.13. Database Management System**

Pengolahan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (Sistem) yang khusus. Perangkat lunak inilah (disebut DBMS/Database Management System) yang akan menentukan bagaimana data di organisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan/konsistensi data dan sebagainya[13].

#### **2.2.14. MySQL**

MySQL merupakan database engine atau server database yang mendukung bahasa database pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

