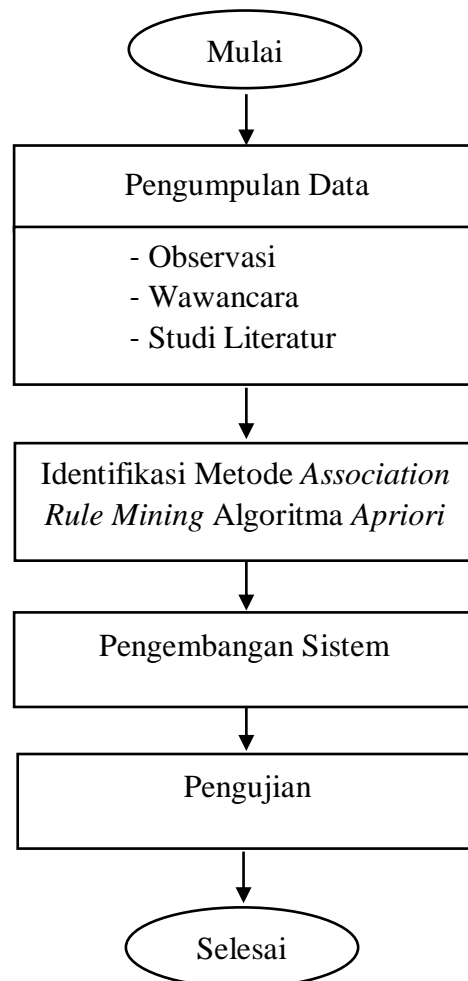


## BAB III

### METODOLOGI

#### 3.1 Metodologi Penelitian

Metodologi penelitian merupakan langkah untuk memperoleh data untuk diproses menjadi informasi yang lebih akurat dan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tinjauan yang akan dicapai. Berikut metodologi penelitian yang digunakan pada penelitian ini dapat dilihat pada gambar 3.1.



**Gambar 3.1 Metodologi Penelitian**

Metodologi penelitian bertujuan untuk menguraikan seluruh kegiatan yang akan dilaksanakan selama kegiatan penelitian berlangsung. Dari gambar diatas, dapat diketahui bahwa ada 4 tahapan yang akan dilakukan untuk menyelesaikan pada penelitian tugas akhir ini meliputi : pengumpulan data melalui observasi, wawancara dan studi literatur, selanjutnya melakukan identifikasi penerapan metode *association rule mining* menggunakan algoritma *apriori* untuk diterapkan pada sistem yang akan dibangun, pengembangan sistem dan pengujian.

## **3.2 Pengumpulan Data**

### **3.2.1 Observasi**

Observasi merupakan pengamatan langsung ke tempat penelitian di PT. Telkomsel *Branch* Tasikmalaya untuk memperoleh informasi yang dibutuhkan dalam penelitian seperti data pendataan *outlet* atau konter seluler dan data hasil pantauan pencapaian berupa data transaksi penjualan produk kartu perdana kuota internet dan pendataan produk lainnya, dengan observasi maka dapat diketahui pokok permasalahan yang dihadapi yang berkaitan dengan sistem serta metode penerapan yang akan dibangun.

### **3.2.2 Wawancara**

Wawancara ini dilakukan untuk mengetahui permasalahan yang sebenarnya terjadi dalam penelitian, proses wawancara dilakukan dengan tanya jawab langsung kepada pihak pimpinan dari PT. Telkomsel *Branch* Tasikmalaya. wawancara yang dilakukan mencakup tentang proses pengolahan data transaksi penjualan berupa produk kartu perdana kuota internet dan data pendukung lainnya.

### 3.2.3 Studi Literatur

Studi Literatur yaitu mempelajari literatur yang berhubungan dengan konsep dari penerapan *Association Rule* dan *Algoritma Apriori*. Sumber literatur banyak di dapatkan dari buku, paper atau jurnal, karya ilmiah, dan situs-situs penunjang lainnya, untuk membantu dalam menyelesaikan tugas akhir serta mengambil dari jurnal penelitian terkait terdahulu.

### 3.3 Identifikasi Metode *Association Rule Mining* Algoritma *Apriori*

Analisis asosiasi atau *association rule mining* adalah teknik data mining untuk menemukan aturan suatu kombinasi *item*. Salah satu tahap analisis asosiasi yang menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi (*frequent pattern mining*). Algoritma *apriori* termasuk jenis aturan asosiasi pada data *mining*. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analysis* atau *market basket analysis* (Pane, Dewi Kartika, 2013).

#### 3.3.1 Proses Asosiasi Algoritma *Apriori*

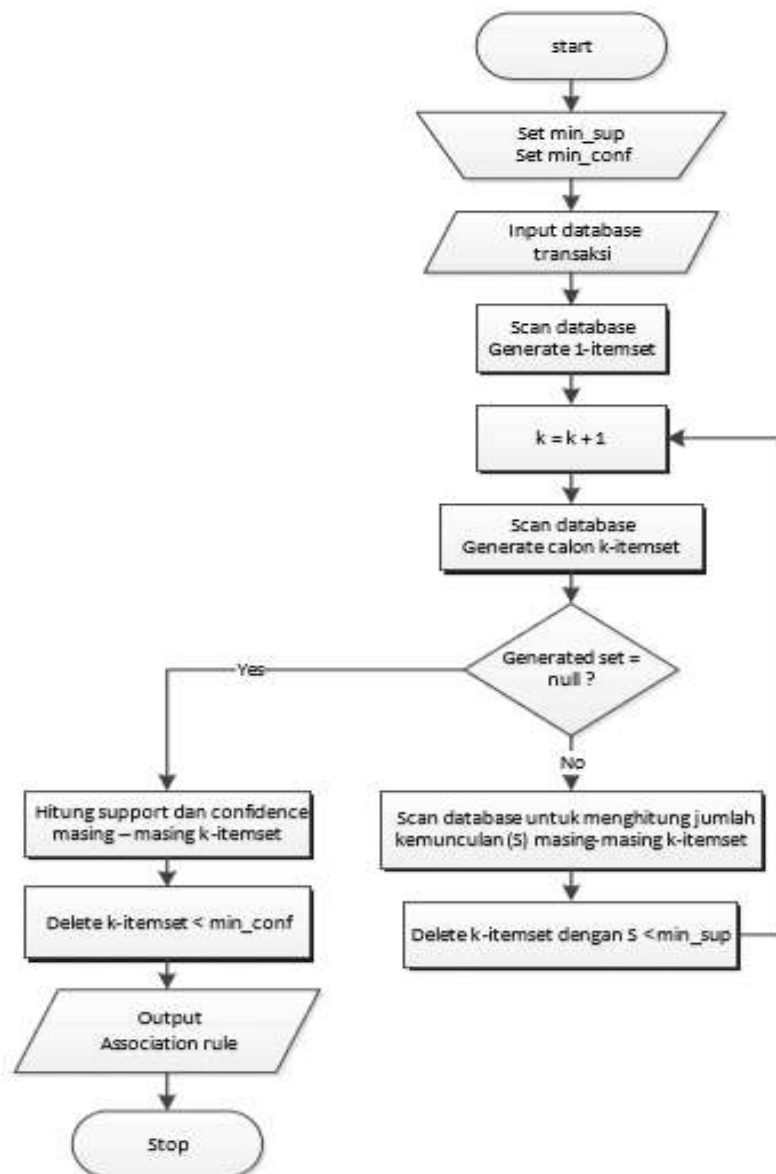
Secara umum dalam pembentukan aturan asosiasi dengan algoritma *apriori* ada dua tahapan yaitu pertama, mencari *frequent itemset* (himpunan item yang memenuhi nilai minimum *support*) yang tersembunyi pada *database* transaksi atau keranjang belanja. Kedua, membentuk aturan asosiasi dari *frequent itemset* yang telah didapat dengan menggunakan nilai minimum *confidence*.

Berikut adalah bagian secara umum dari atribut algoritma apriori (Gama et al, 2016):

- a. Algoritma *apriori* menggunakan pendekatan iteratif dimana *k-itemset* digunakan untuk mengeksplorasi *(k-1)-itemset*.
- b. Calon *(k+1)-itemset* dengan frekuensi yang jarang muncul atau dibawah *threshold (min\_support)* akan dipangkas dan tidak dipakai dalam menentukan aturan asosiasi.
- c. *1-itemset* ditemukan dengan melakukan *scanning database* untuk mengakumulasi jumlah dari setiap kategori *item* dan kemunculannya pada setiap transaksi.
- d. Selanjutnya *1-itemset* digunakan untuk menemukan *2-itemset*. Calon kandidat *2-itemset* ditemukan dengan cara saling memasang satu *item* dengan item yang lainnya sehingga didapat kombinasi yang memungkinkan untuk 2 buah item.
- e. *2-itemset* kemudian dihitung nilai kemunculannya pada setiap transaksi. Nilai ambang (*min\_support*) ditentukan untuk memangkas calon kandidat yang dianggap tidak sering muncul atau tidak *frequent*.
- f. *2-itemset* yang memenuhi nilai ambang akan dihitung nilai *support* dan *confidence*-nya. *2-itemset* yang memenuhi *min\_support* dan *min\_confidence* akan dijadikan aturan asosiasi.
- g. *2-itemset* digunakan untuk menemukan *3-itemset* dan seterusnya sampai tidak ada lagi *frequent (k+1)-itemset* yang bisa ditemukan.

- h. Setelah semua aturan dari *frequent (k+1)-itemset* terbentuk, kemudian dihitung nilai *support* dan *confidence*. Hasil perkalian dari nilai *support* dan *confidence* yang paling tinggi merupakan aturan asosiasi yang paling baik dari keseluruhan transaksi yang ada dalam database.

*Flowchart* atau bagan alir dari algoritma *apriori* yang diimplementasikan pada *database* transaksi untuk menemukan *frequent itemset* ditunjukkan pada Gambar 3.2.



**Gambar 3.2** *Flowchart* Algoritma *Apriori* Gama et al (2016).

### 3.3.2 Tahapan Algoritma *Apriori* pada Aturan *Association Rule*

Tahapan Algoritma *Apriori* pada Aturan *Association Rule* yaitu:

- a. Transformasi data dalam bentuk tabel tabular.
- b. Menentukan nilai minimal *support* dan minimal *confidence*.

- c. Pembentukan kandidat 1-*itemset* 1(satu) pola kombinasi kemudian hitung jumlah kemunculannya pada setiap *itemset*.
- d. Pembentukan kandidat 2-*itemset*, kombinasi dari 2-*itemset* data untuk semua *item* hingga tidak bisa lagi dikombinasikan (disebut calon kombinasi 2-*itemset* atau calon F2).
- e. Pemangkasan atau pembuangan *itemset* yang memiliki nilai  $<$  minimum *support* (yang diterima adalah frekuensi item set  $\geq$  minimum *support*). Hal ini dilakukan untuk menemukan F2 akhir.
- f. Setelah menemukan F2 akhir, kemudian melanjutkan untuk menemukan kembali kombinasi 3-*itemset*. Buat kombinasi 3-*itemset* untuk semua *item* sehingga tidak bisa lagi dikombinasikan.
- g. Jika tidak ada lagi *item* yang dapat dikombinasikan maka hitung nilai *support* dan *confidence* untuk masing- masing F2 dan F3.
- h. Nilai *support* diperoleh dengan rumus yang digunakan untuk kombinasi 2-*itemset* dan seterusnya:

$$Support (A \cap B) = \frac{Jumlah\ transaksi\ menggunakan\ A\ dan\ B}{Total\ Transaksi} \times 100\% \quad (3.1)$$

$$Support (A \cap B \cap C) = \frac{Jumlah\ kejadian\ menggunakan\ A,B\ dan\ C}{Total\ kejadian} \times 100\% \quad (3.2)$$

- i. Nilai *confidence* diperoleh dengan rumus yang digunakan untuk kombinasi 2-*itemset* dan seterusnya:

$$Confidence (A/B) = \frac{Jumlah\ transaksi\ menggunakan\ A\ dan\ B}{Total\ Transaksi\ mengandung\ A} \times 100\% \quad (3.3)$$

$$Confidence (A/B/C) = \frac{Jumlah\ transaksi\ menggunakan\ A,B\ dan\ C}{Total\ Transaksi\ mengandung\ A} \times 100\% \quad (3.4)$$

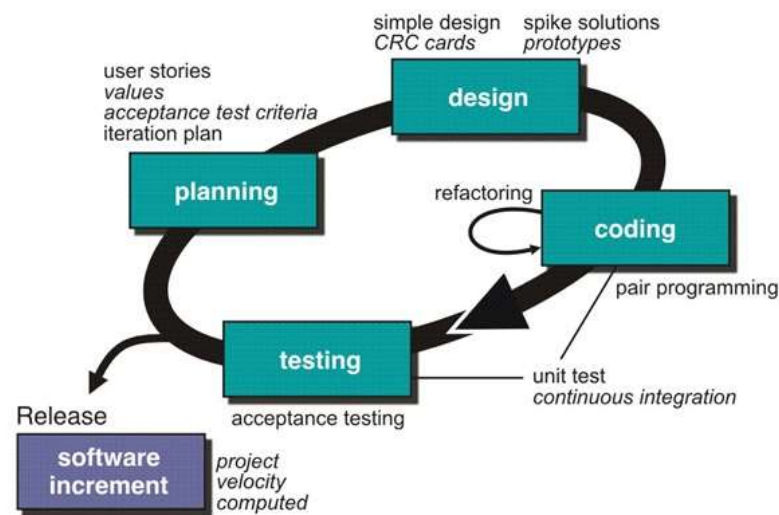
- j. Selanjutnya menentukan nilai minimum *confidence*, lalu pangkas *itemset* yang kurang dari nilai minimum *confidence* (yang diterima adalah nilai *confidence*  $\geq$  minimum *confidence*).
- k. Setelah didapat nilai *Support* dan *Confidence* untuk masing-masing kandidat lakukan perkalian antara *Support* dan *Confidence*.
- l. Setelah didapat hasil perkalian antara nilai *Support* dan *Confidence* pilihlah yang hasil perkaliannya paling besar. Hasil paling besar dari perkalian - perkalian tersebut merupakan *rule* yang di pakai dalam aturan Asosiasi final.
- m. Analisis Asosiasi Final digunakan untuk menganalisis hasil dari Algoritma *Apriori*. Berdasarkan informasi yang telah disampaikan terdapat 2 buah penilaian yaitu *Support* dan *Confidence*. *Support* digunakan untuk mengukur kemungkinan sebuah *itemset* diambil atau digunakan secara bersamaan, sedangkan *Confidence* digunakan untuk mengukur kepastian hubungan antar *itemset*.

### 3.4 Pengembangan Sistem

Pengembangan sistem merupakan tahapan dalam membangun sebuah sistem yang dibangun untuk kebutuhan data informasi dalam melakukan proses pengolahan data dengan menerapkan metode *association rule mining* menggunakan perhitungan algoritma *apriori*. Model proses pengembangan perangkat lunak yang digunakan adalah *Extreme Programming (XP)* merupakan



suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Alasan menggunakan metode *Extreme Programming (XP)* karena sifat dari aplikasi yang di kembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi : *Planning* (Perencanaan), *Design* (Perancangan), *Coding* (Pengkodean), dan *Testing* (Pressman, 2012:88). Adapun tahapan pada *Extreme Programming* dapat di jelaskan sebagai berikut:



**Gambar 3.3 Metode Pengembangan *Extreme Programming* (Pressman, 2012:88)**

#### 1. *Planning* (Perencanaan)

Pada tahap perencanaan ini dimulai dari pengumpulan kebutuhan yang membantu tim teknikal untuk memahami konteks bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikan output yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang dikembangkan.

## 2. *Design* (Perancangan)

Tahap ini merupakan perancangan terhadap pemecahan masalah yang akan dibangun. Pada tahapan ini dilakukan kegiatan pembuatan Diagram Konteks, *Data Flow Diagram* (DFD) dan desain antar muka.

## 3. *Coding* (Pengkodean)

Tahap pengkodean merupakan tahapan dimana keseluruhan desain diubah menjadi kode-kode program. Implementasi Bahasa pemrograman yang akan digunakan untuk perancangan Aplikasi ini adalah berbasis *object oriented / java*, PHP dan *database* menggunakan MySQL.

## 4. *Testing* (Pengujian)

Setelah selesai di bangun dengan pemrograman, selanjutnya dilakukan proses pengujian sistem dengan menggunakan metode *Black Box Testing*, dilakukan oleh user.

### **3.5 Pengujian**

Pengujian merupakan tahapan dimana aplikasi akan dijalankan. Tahap pengujian dilakukan untuk menguji dari aplikasi yang telah dibuat. Dalam penelitian ini pengujian dilakukan untuk menguji mengenai penerapan dari metode *association rule mining* menggunakan perhitungan algoritma *apriori*. Pengujian ini bertujuan untuk membandingkan perhitungan yang dilakukan antara perhitungan manual dengan perhitungan sistem yang telah dibangun, hal ini berguna untuk mengetahui bahwa perhitungan yang dilakukan oleh sistem sudah sesuai dengan konsep perhitungan algoritma *apriori* itu sendiri.