

## **BAB III**

### **METODOLOGI**

Pada bab ini akan dibahas data dan metodologi yang akan digunakan pada penelitian ini. Penjelasan terdiri dari pembahasan data dan metodologi yang digunakan.

#### **3.1 DATA**

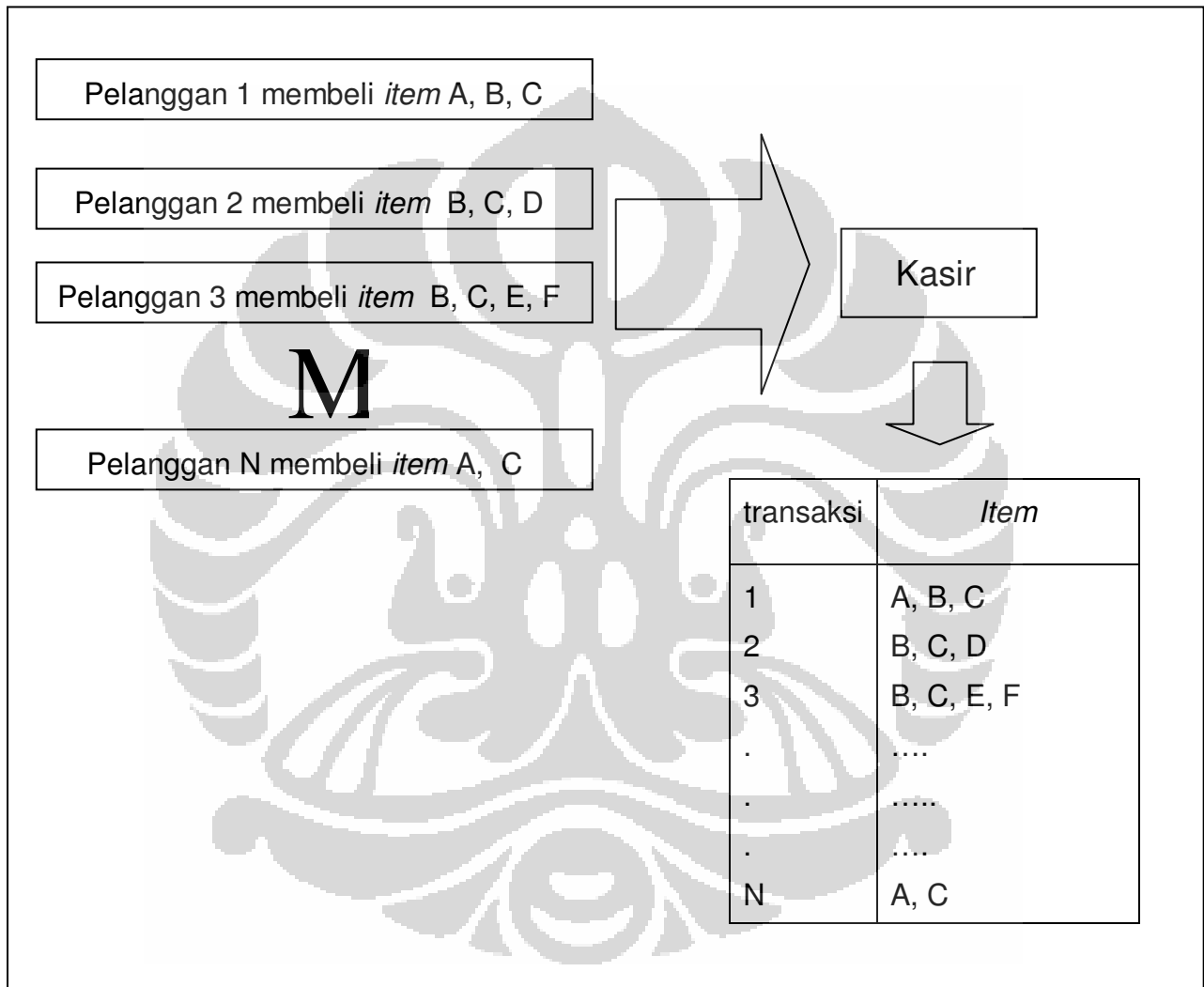
Data yang digunakan dalam penelitian ini adalah data transaksi penjualan. Data transaksi penjualan berisi informasi dari setiap transaksi penjualan jenis produk yang dijual oleh suatu minimarket. Informasi yang dimaksud adalah jenis produk yang terjual untuk setiap transaksinya.

#### **3.2. PENGAMBILAN SAMPEL**

Sampel diambil di salah satu minimarket di daerah Perumnas Klender. Data diambil dengan cara mencatat jenis-jenis produk yang dibeli oleh tiap pelanggan melalui struk yang dikeluarkan oleh kasir. Bentuk data yang didapat dapat dilihat pada lampiran 1.

Proses pengambilan data dapat di jelaskan pada contoh di Gambar 3.1

berikut.:



**Gambar 3.1** . Proses Pengambilan Data

### 3.3 METODE KAJIDAH ASOSIASI

Metode Kaidah Asosiasi adalah metode untuk menghasilkan aturan-aturan asosiasi sehingga dapat diketahui produk-produk apa saja yang sering dibeli secara bersamaan oleh konsumen.

### 3.4 ATURAN ASOSIASI

Aturan asosiasi adalah aturan yang digunakan untuk mengetahui *item-item* yang berasosiasi. Aturan-aturan asosiasi berbentuk jika “kejadian sebelumnya” maka “konsekuensinya”. Misalnya “jika seorang membeli roti maka ia membeli susu”. Aturan-aturan asosiasi ini nantinya akan memberikan informasi tentang kombinasi *item* yang dibeli secara bersamaan oleh pelanggan.

Kuat atau tidaknya sebuah aturan asosiasi ditentukan oleh dua parameter yaitu *support* dan *confidence*.

Misalkan A dan B adalah *itemset*,  $\lambda = \{\text{indeks pada A}\}$  dan  $\beta = \{\text{indeks pada B}\}$ .

### 3.4.1 Support

*Support* dari suatu aturan ( $A \Rightarrow B$ ) adalah proporsi kejadian di mana *item-item* di  $A$  dan  $B$  terbeli secara bersamaan . Dilambangkan dengan  $S(A \Rightarrow B)$ . Berdasarkan Teorema 1 pada Lampiran 3, maka  $S(A \Rightarrow B)$  dirumuskan sebagai berikut:

$$S(A \Rightarrow B) = \frac{1}{N} \sum_{i=1}^N \prod_{p \in \lambda \cup \beta} x_{ip} \quad (3)$$

Intepretasi : “kemungkinan seorang pelanggan membeli *item* di himpunan  $A$  dan  $B$  sebesar  $S(A \Rightarrow B)$ ”.

### 3.4.2 Confidence

*Confidence* dari suatu aturan ( $A \Rightarrow B$ ) adalah ukuran akurasi dari suatu aturan asosiasi. Dilambangkan dengan  $C(A \Rightarrow B)$ . Rumus dari  $C(A \Rightarrow B)$  adalah:

$$C(A \Rightarrow B) = \frac{\frac{1}{N} \sum_{i=1}^N \prod_{p \in (\lambda \cup \beta)} x_{ip}}{\frac{1}{N} \sum_{i=1}^N \prod_{a \in \lambda} x_{ia}} = \frac{\sum_{i=1}^N \prod_{p \in (\lambda \cup \beta)} x_{ip}}{\sum_{i=1}^N \prod_{a \in \lambda} x_{ia}} \quad (4)$$

Intepretasi : “Jika seorang pelanggan membeli produk-produk di himpunan A, maka kemungkinan sebesar  $C(A \Rightarrow B)$  pelanggan itu akan membeli produk-produk di himpunan B juga”.

### Definisi 8

Suatu aturan asosiasi ( $A \Rightarrow B$ ) disebut kuat jika memiliki *confidence* lebih besar dari *minimum confidence*

$$C(A \Rightarrow B) = \frac{\sum_{i=1}^N \prod_{p \in (\lambda \cup \beta)} x_{ip}}{\sum_{i=1}^N \prod_{a \in \lambda} x_{ia}} \geq \text{confidence minimum} \quad (5)$$

Untuk mendapatkan aturan-aturan asosiasi seperti di atas, diperlukan suatu algoritma. Salah satu algoritma yang populer digunakan adalah algoritma Apriori.

### 3.5. ALGORITMA APRIORI

Dengan diberikan nilai *minimum support* dan *minimum confidence*, pencarian aturan-aturan asosiasi dengan menggunakan algoritma Apriori ada dua tahap:

1. Temukan *itemset-itemset* yang *frequent*.
2. Bangkitkan aturan asosiasi dari setiap *itemset* yang *frequent*.

Arti Apriori secara umum adalah anggapan atau sikap yang sudah ditentukan sebelum mengetahui (melihat, menyelidiki) terhadap sesuatu.

Ide dari algoritma Apriori adalah mengurangi jumlah kandidat-kandidat tersebut yang harus diperiksa dengan jaminan properti dari Apriori.

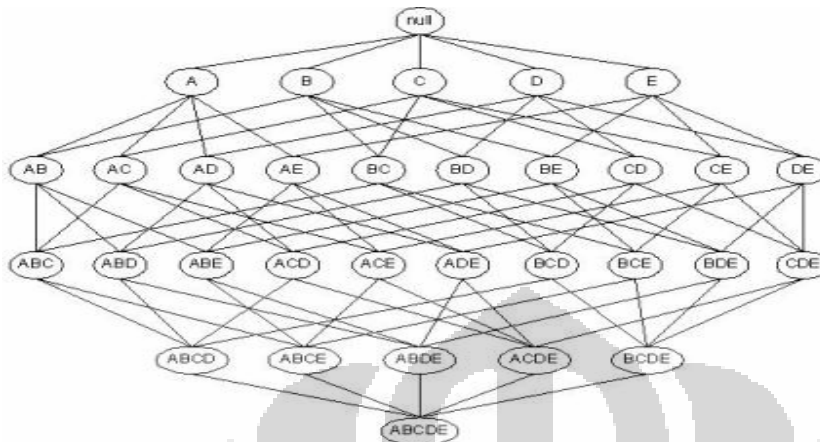
#### **Properti Apriori:**

1. Dalam menghasilkan sebuah *frequent itemset* dibutuhkan satu kali *scan* database transaksi.
2. Jika suatu *itemset* A memiliki *support* kurang dari *minimum support*, maka *itemset* tersebut tidak *frequent* (*infrequent*).
3. Seluruh *nonempty subset* dari suatu *frequent itemset* juga harus *frequent*. (Bukti di Teorema 2 pada Lampiran 3).

#### **3.5.1 Pencarian *Itemset-Itemset* yang *Frequent***

Masalah utama pencarian dalam *itemset* yang *frequent* adalah banyaknya jumlah kombinasi *itemset* yang harus diperiksa apakah memenuhi *minimum support* atau tidak. Salah satu cara untuk mengatasinya adalah dengan mengurangi jumlah kandidat untuk *frequent itemset* yang harus diperiksa.

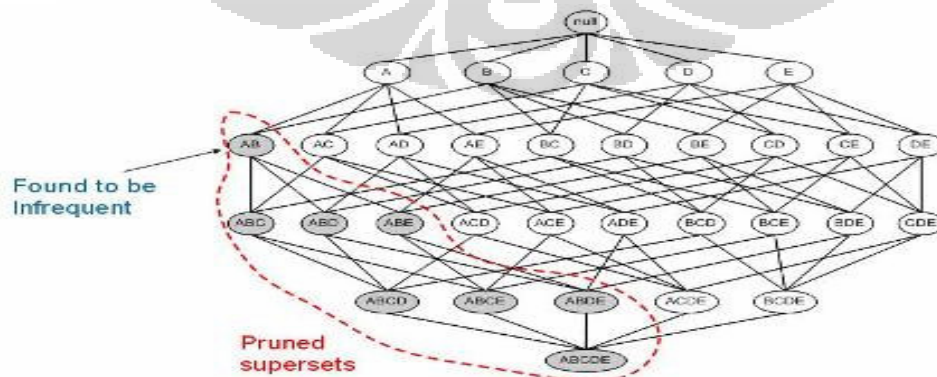
Untuk lebih jelasnya dapat dijelaskan melalui Gambar 3.2 berikut:



**Gambar 3.2** Pencarian Kandidat *Itemset* Tanpa Menggunakan Algoritma Apriori

Pada Gambar 3.2 di atas, pencarian *itemset* yang *frequent* dilakukan tanpa menggunakan prinsip Apriori. Setiap kandidat dari  $k$ -*itemset* harus di scan lalu dicari *supportnya*. Hal ini akan memakan waktu yang banyak apabila dilakukan pada jenis *item* dan transaksi yang besar.

Dengan menggunakan prinsip Apriori, pencarian *itemset* yang *frequent* akan menjadi seperti Gambar 3.3 di bawah ini:



**Gambar 3.3** Pencarian Kandidat *Itemset* dengan Menggunakan Algoritma Apriori

Dapat dilihat bahwa dengan menggunakan Apriori, jumlah kandidat yang harus diperiksa cukup banyak berkurang. Hal ini dikarenakan properti Apriori menjamin bahwa jika suatu *itemset* tidak *frequent* maka semua *superset*-nya tidak *frequent*.

Berikut adalah implementasi dari algoritma Apriori :

Algoritma Apriori

$C_k = \{\text{kandidat } k\text{-itemset}\}$   
 $F_k = \{\{k\text{-itemset yang frequent}\}$

*Join Step* :  $C_k$  dibentuk dari  $F_{k-1}$  join  $F_{k-1}$   
*Prune Step*: Setiap  $k$ -itemset yang tidak *frequent* memiliki *superset* yang tidak *frequent*

$F_1 = \{1\text{-itemset frequent}\}$

**For** (  $k=2$ ;  $F_{k-1} \neq \emptyset$ ;  $k++$  ) **do begin**  
     Bentuk  $C_k$  dari  $F_{k-1}$  join  $F_{k-1}$  (*Join Step*)  
     Pangkas anggota dari  $C_k$  yang memiliki *subset* yang tak *frequent* (*Prune Step*)  
     Untuk setiap  $k$ -itemset  $A$  anggota  $C_k$  yang tak memiliki *subset* yang tak *frequent*, cari *support* ( $A$ ).  
      $F_k = \{A \mid \text{support}(A) \geq \text{minimum support}\}$

**end**  
**return**  $\cup_k F_k$ ,



### 3.5.2. Bentuk Aturan Asosiasi

Setelah didapat semua *itemset* yang *frequent*, bentuk aturan-aturan asosiasi untuk setiap *itemset* yang *frequent* dengan prosedur :

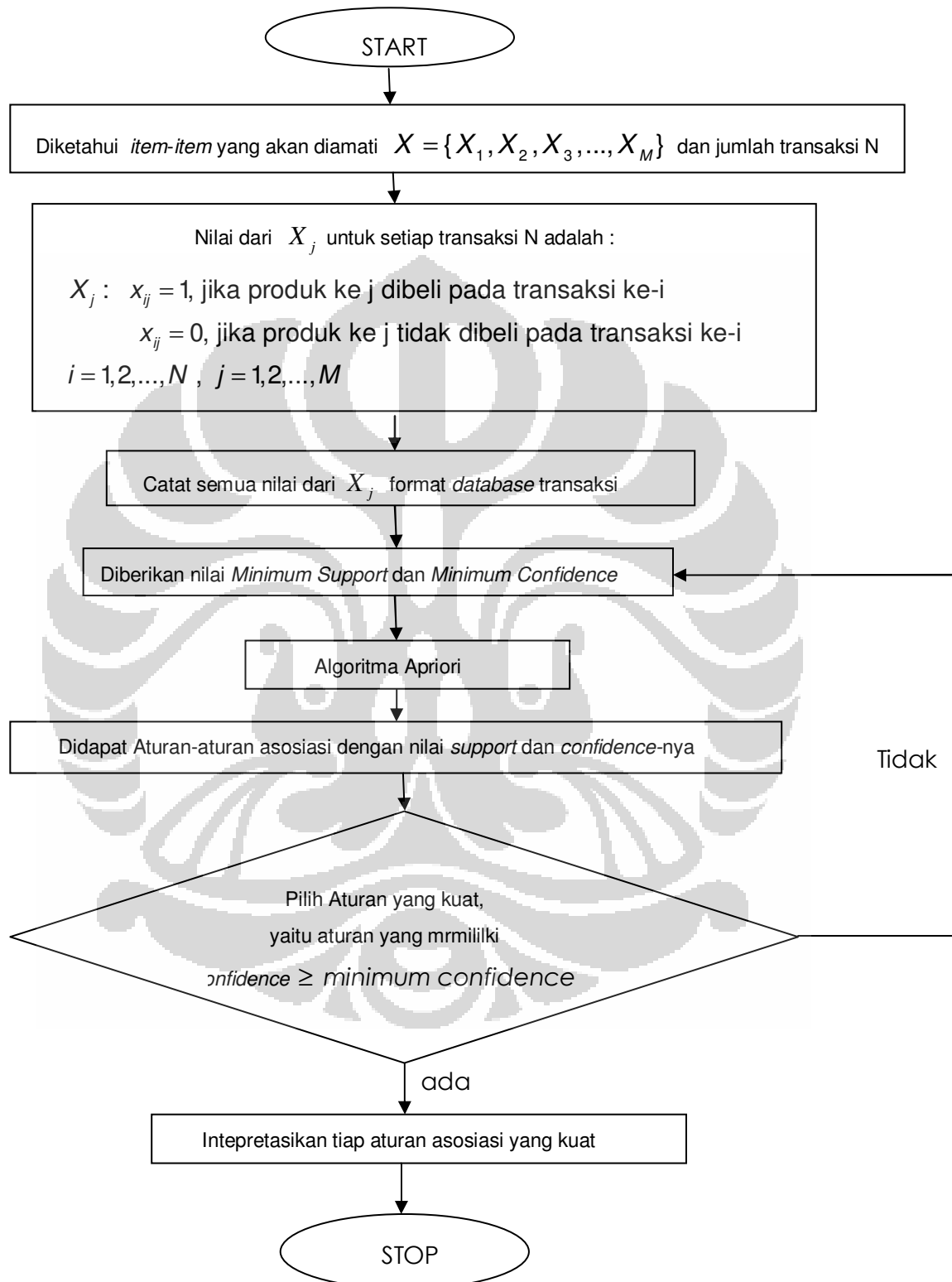
1. Untuk setiap *frequent itemset*  $I$  bangkitkan semua *nonempty subset* dari  $I$
2. Untuk setiap *nonempty subset*  $S$  dari  $I$  *output* aturannya adalah " $S \Rightarrow (I - S)$ " dengan  $I - S \neq \emptyset$  beserta nilai *support* dan *confidence*nya

Setelah didapat semua aturan-aturan asosiasi, pilih aturan-aturan yang kuat, yaitu aturan yang memiliki *confidence* lebih besar dari *minimum confidence*.

### 3.6. INTERPRETASI ATURAN ASOSIASI

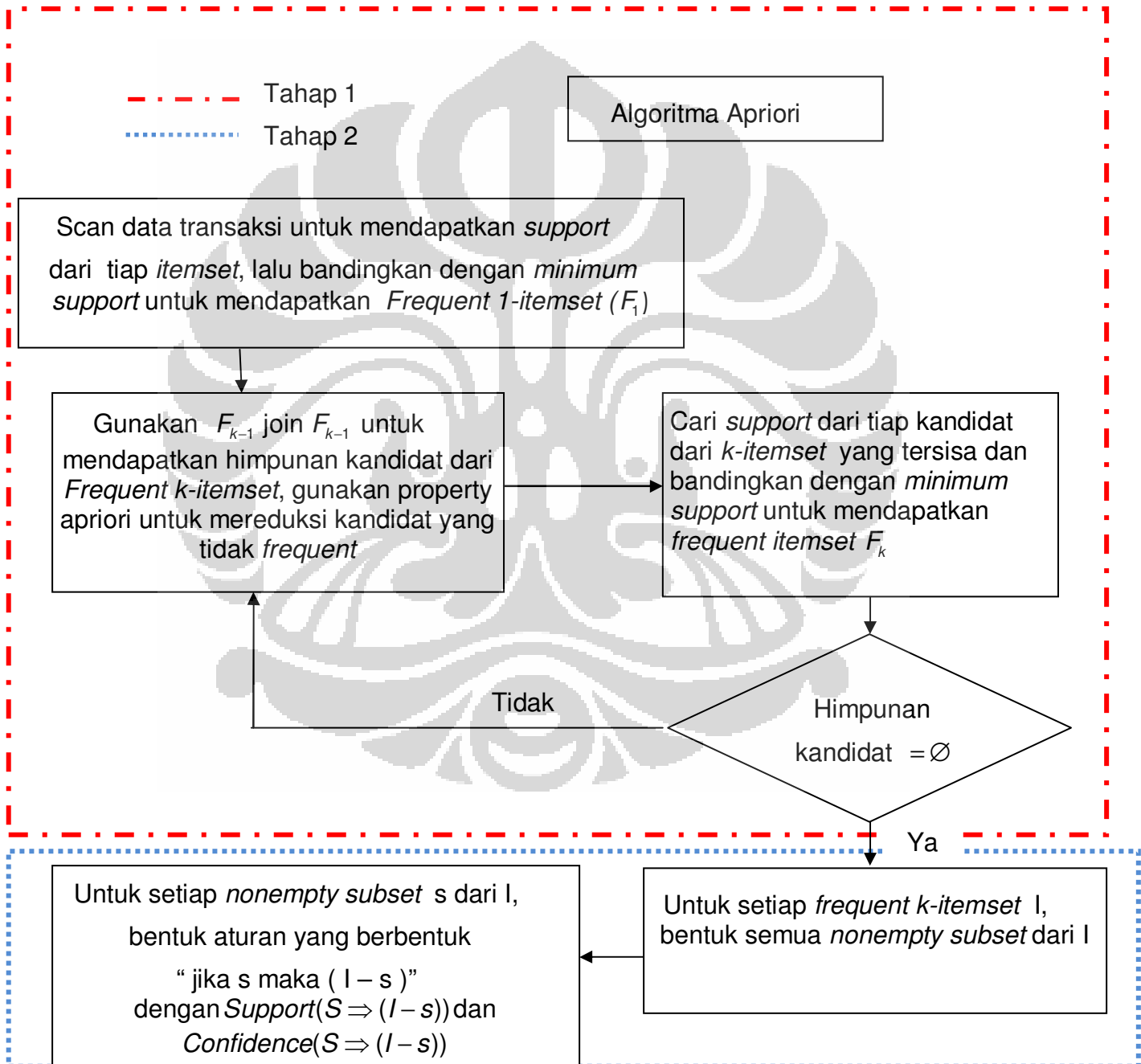
Intepretasi dari aturan asosiasi jika A maka B dengan *support* =  $S(A \Rightarrow B)$  dan *confidence* =  $C(A \Rightarrow B)$  adalah "Jika seorang konsumen membeli produk-produk di himpunan A, maka dengan kemungkinan  $C(A \Rightarrow B)$  ia akan membeli produk-produk yang ada di B juga, hal ini cukup signifikan karena mewakili  $S(A \Rightarrow B)$  dari database yang ada".

Untuk lebih jelasnya dapat dilihat dalam diagram alir metode kaidah asosiasi pada Gambar 3.3 berikut :



**Gambar 3.3.** Diagram Alir Metode Kaidah Asosiasi

Dalam diagram alir pada Gambar 3.3 terdapat step algoritma Apriori, step tersebut dapat dijelaskan melalui diagram alir algoritma Apriori pada Gambar 3.4 berikut:



**Gambar 3.4.** Diagram Alir Algoritma Apriori