

BAB III

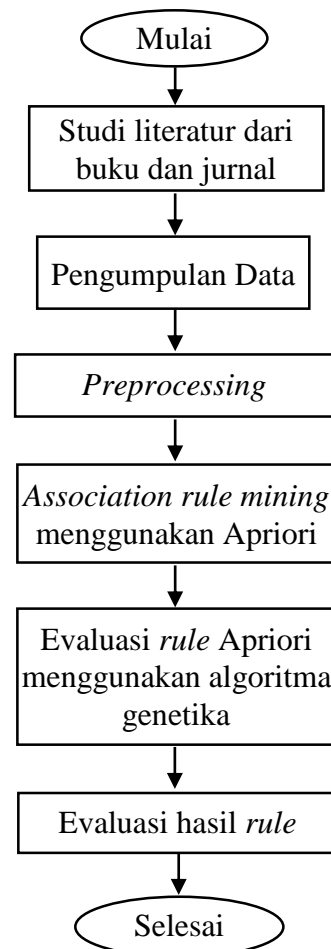
METODOLOGI PENELITIAN

1.1. Objek Penelitian

Pada penelitian ini, objek penelitian yang digunakan adalah *dataset* dari UCI Machine Learning Repository dengan subjek Online Retail Data Set. *Dataset* ini memiliki delapan atribut. *Dataset* ini diakses dari *link* <http://archive.ics.uci.edu/ml/datasets/online+retail>.

1.2. Diagram Alir Penelitian

Pada bab ini akan dijelaskan tahapan penelitian yang diterapkan seperti yang ditunjukkan pada Gambar 3.1



Gambar 3.1 Alur Metode Penelitian

1.3. Studi Literatur

Peneliti melakukan studi literatur pada penelitian sebelumnya yang berkaitan dengan topik penelitian yang dilakukan sebagai sumber referensi bagi peneliti. Sumber studi literatur diambil dari buku, penelitian survey dan *technical paper*. Penelitian terkait yang dijadikan rujukan utama dalam penelitian ini adalah penelitian Shantha Rangaswamy dan Ahobha G. [8], Penelitian Goldie Gunadi dan Dana Indra Sensuse [9] dan penelitian M. Dhanabhakym and M. Punithavalli [3].

1.4. Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan *dataset* dari UCI *Machine Learning Repository* dengan subjek *Online Retail Data Set*. *Dataset* ini dibagikan oleh Dr. Daqing Chen yang merupakan seorang dosen senior di London South Bank University. *Online Retail Data Set* merupakan *dataset* transaksional yang mengandung 541.909 jumlah baris dan semua transaksi yang terjadi diantara tanggal 1 Desember 2010 sampai 9 Desember 2011 pada retail *online non-store* yang berbasis di United Kingdom. Semua atribut *dataset* dan sampel data ditunjukkan pada Tabel 3.1.

Tabel 3.1 Sampel *Dataset* ‘Online Retail Data Set’

Atribut	Dataset				
InvoiceNo	536365	536365	536365	...	581587
Stock Code	85123A	71053	84406B	...	22138
Description	WHITE HANGING HEART T- LIGHT HOLDER	WHITE METAL LANTERN	CREAM CUPID HEARTS COAT HANGER	...	BAKING SET 9 PIECE RETROSPOT
Quantity	6	6	8	...	3
Invoice Date	01/12/2010 8:26:00	01/12/2010 8:26:00	01/12/2010 8:26:00	...	09/12/2011 12:50:00
UnitPrice	2,55	3,39	2,75	...	4,95
CustomerID	17850	17850	17850	...	12680

Country	United Kingdom	United Kingdom	United Kingdom	...	France
---------	----------------	----------------	----------------	-----	--------

Ada beberapa atribut yang terdapat dalam *dataset* ini, yaitu :

- a. InvoiceNo, merupakan nomor transaksi/*invoice* yang terdiri dari enam digit angka yang bersifat unik untuk setiap transaksi. Jika nomor transaksi dimulai dengan huruf 'c', maka transaksi terindikasi dibatalkan
- b. StockCode, merupakan kode produk/item terdiri dari lima digit angka yang bersifat unik untuk setiap item
- c. Description, nama item
- d. Quantity, kuantitas setiap item yang dibeli per transaksi
- e. InvoiceDate, merupakan tanggal dan waktu ketika transaksi dilakukan
- f. UnitPrice, harga item per unit dalam mata uang
- g. CustomerID, kode untuk pelanggan terdiri dari lima digit angka yang bersifat unik untuk setiap pelanggan
- h. Country, nama negara dimana setiap pelanggan menetap

1.5. *Pre-processing*

Tahap *preprocessing* dilakukan secara manual untuk menyeleksi dan memvalidasi *dataset* dari data yang memiliki sifat *noise*, duplikasi data, atribut yang hilang dan data yang tidak sesuai dengan format yang ditetapkan. *Preprocessing* mencegah kualitas data rendah yang akan mempengaruhi tingkat akurasi pada proses *mining*.

Kondisi awal pada *dataset* yang digunakan memiliki delapan atribut dan 541.909 baris data. Untuk mendapatkan data yang bagus dan sesuai dengan format untuk *market basket analysis*, berikut adalah tahapan *preprocessing* yang dilakukan dalam penelitian ini :

a. *Data Cleaning*

Data cleaning dilakukan dengan menghapus *missing value* dan inkonsistensi data pada atribut InvoiceNo. Inkonsistensi data yang dihilangkan adalah InvoiceNo yang memiliki kode 'c' yang berarti *cancellation* atau pembatalan transaksi. Langkah selanjutnya dengan

menghapus data InvoiceNo yang memiliki duplikasi sehingga setiap InvoiceNo bersifat unik.

b. *Data Reduction*

Semua atribut pada *dataset* tidak digunakan untuk proses selanjutnya. Oleh karena itu *data reduction* dilakukan untuk menghapus atribut yang tidak digunakan. Pada *dataset* ini atribut yang digunakan hanya InvoiceNo dan StockCode. Sementara atribut Description, Quantity, InvoiceDate, UnitPrice, CustomerID dan Country akan dihapus.

c. *Data Transformation*

Pada data keranjang belanja (*market basket data*), data yang valid adalah data setiap transaksi pada setiap baris/objek dan *item* sebagai atributnya. Dalam *dataset* yang digunakan, yang menjadi atribut adalah setiap anggota dari StockCode yang bersifat unik. Untuk itu, setiap duplikasi data dan data yang tidak valid dihapus dari StockCode dan dijadikan sebagai atribut. Nilai setiap data adalah 1 atau 0 yang berarti pada transaksi ke-*n*, *item* ke-*n* dibeli (1) dan pada transaksi ke-*n*, *item* ke-*n* tidak dibeli (0). Gambaran *dataset* setelah proses *data transformation* ini ditunjukkan pada tabel 3.2.

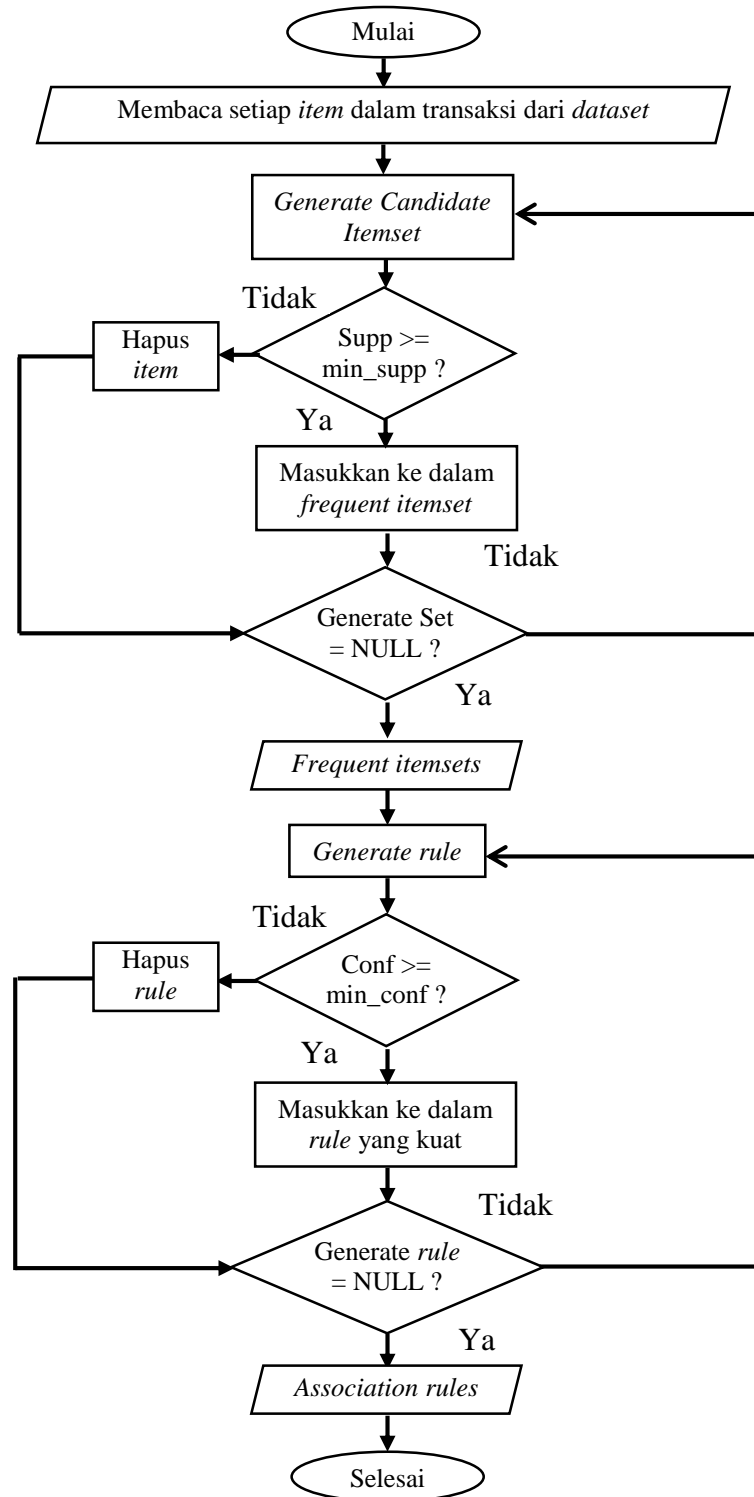
Tabel 3.2 *Dataset Setelah Preprocessing*

InvoiceNo	85123A	71503	84406B	84029E	22752	...	23843
536365	1	1	1	1	1	...	0
536366	0	0	0	0	0	...	0
536367	0	0	0	0	0	...	0
...
581587	0	0	0	0	0	...	0

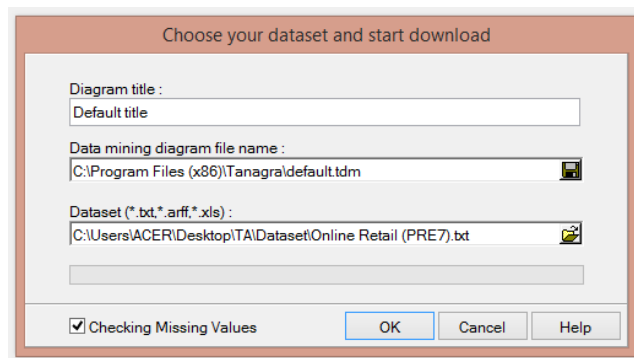
Setelah *dataset* melalui proses *preprocessing* hingga mempunyai format seperti pada Tabel 3.2, *dataset* sekarang mempunyai 3930 StockCode sebagai jumlah atribut dan 21870 baris data.

1.6. Association Rule Mining Menggunakan Apriori

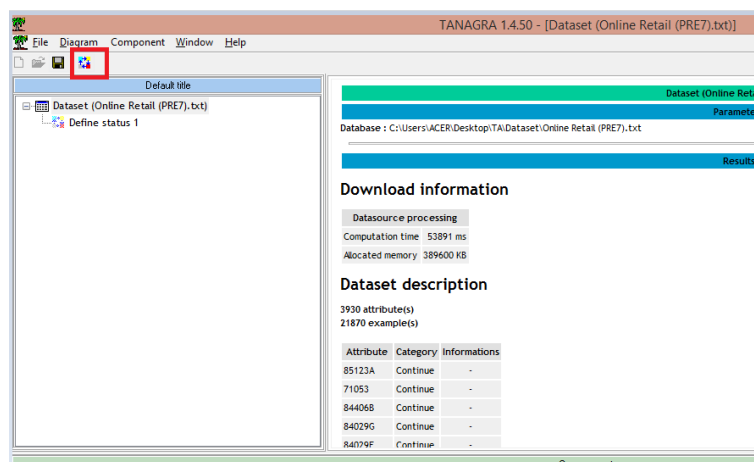
Langkah-langkah dalam algoritma apriori ditunjukkan pada Gambar 3.2.



Gambar 3.2 Tahapan Algoritma Apriori



Gambar 3.4 Proses Memasukkan *Dataset* ke Dalam TANAGRA

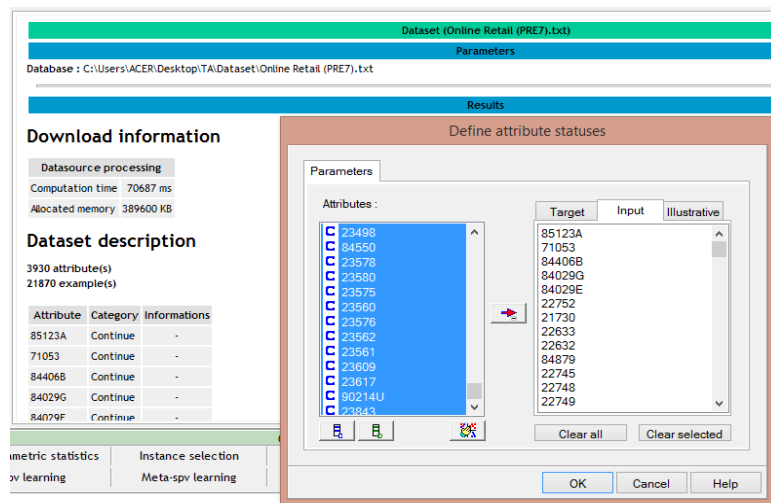


Gambar 3.5 Tampilan Aplikasi TANAGRA Setelah Membaca Sebuah *Dataset*

Dataset yang telah terbaca ditunjukkan pada Gambar 3.5 yang mencakup informasi seperti *computation time*, *allocated memory* dan deskripsi *dataset*. *Dataset* yang dimasukkan adalah data keranjang belanja dengan 3930 atribut yang bersifat continue dan 21870 baris data.

b. Mendefinisikan status atribut

Definisi status atribut dilakukan sebelum melakukan proses *mining* untuk mendefinisikan setiap atribut pada *dataset*. Pada Gambar 3.5, proses ini dilakukan dengan memilih tombol yang ditandai dengan kotak merah. Proses ini ditunjukkan pada Gambar 3.6.

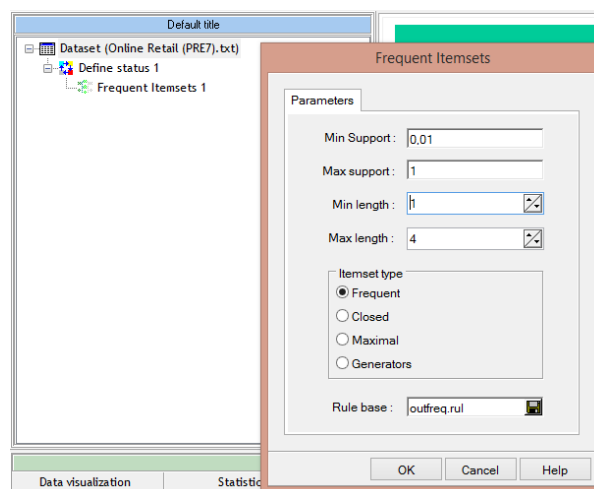


Gambar 3.6 Proses Definisi Status Atribut

Semua atribut yang bersifat *continue* (ditandai dengan kode C) dipilih, kemudian seluruh atribut ini didefinisikan sebagai *Input*.

c. Menentukan *frequent itemset*

Frequent itemset merupakan sebuah populasi yang menunjukkan seberapa sering sebuah *item* atau sekumpulan *item* muncul dalam sebuah transaksi yang nilainya ditentukan oleh nilai *support*. Untuk mencari *frequent itemset* pada TANAGRA ada pada menu *Association* -> *Frequent Itemsets*. Sebelum eksekusi perintah, beberapa parameter yang harus diatur adalah *Min Support*, *Max Support*, *Min Length* dan *Max Length*. Pengaturan parameter ini ditunjukkan pada Gambar 3.7.



Gambar 3.7 Pengaturan Parameter Untuk Pencarian *Frequent Itemset*

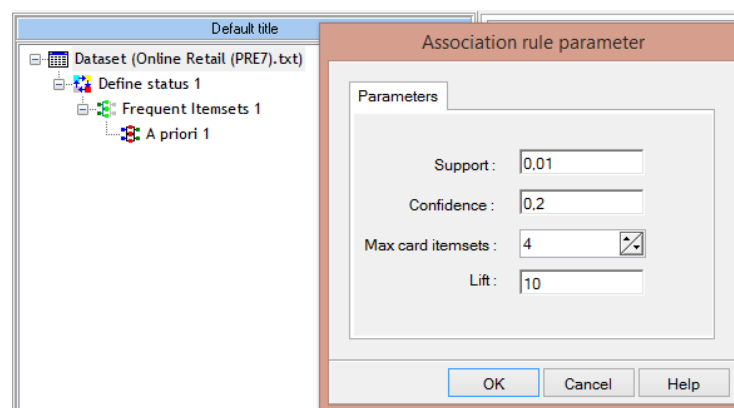
Pada Gambar 3.7, *Min Support* dan *Max Support* menandakan *range* yang harus dipenuhi oleh satu *itemset* agar dapat ditetapkan sebagai *frequent itemset*. *Min Length* dan *Max Length* adalah parameter yang menentukan jumlah *item* yang dihitung pada setiap *itemset*. Hasil dari *frequent itemset* ditunjukkan pada Gambar 3.8

Execution log...		
C:\Program Files (x86)\Tanagra\exe\apriori.exe - find frequent item sets with the apriori algorithm		
version 5.57 (2011.09.02) (c) 1996-2011 Christian Borgelt		
reading C:\Users\ACER\AppData\Local\Temp\dataE9B4.tmp ... [3929 item(s), 21469 transaction(s)] done [0.25s].		
filtering, sorting and recoding items ... [728 item(s)] done [0.01s].		
sorting and reducing transactions ... [17229/21469 transaction(s)] done [0.07s].		
building transaction tree ... [20477 node(s)] done [0.01s].		
checking subsets of size 1 2 3 4 done [0.16s].		
writing C:\Users\ACER\Desktop\ITA\Dataset\outfreq.rul ... [1539 set(s)] done [0.06s].		
ITEMSETS [#1539 itemsets loaded]		
N*	Description	Support
1	21499	1,0
2	23230	1,1
3	21497	1,1
4	23232	1,1
5	16161P	1,1
6	23231	1,2
7	23146	1,0
8	21623	1,0
9	21498	1,5
10	22634	1,0
11	21124	1,1
12	23147	1,2
13	22796	1,1

Gambar 3.8 Hasil *Generate Frequent Itemset*

d. Melakukan *generate rule*

Association rule merupakan keluaran dari apriori yang menentukan seberapa hubungan antara satu *item* dengan *item* yang lain. Untuk menghasilkan *rule* di TANAGRA menggunakan fungsi apriori pada menu *Association* -> *Apriori*. Beberapa parameter yang harus diatur adalah nilai *support*, *confidence*, *max card itemset* dan nilai *lift*. Pengaturan pada TANAGRA ditunjukkan pada Gambar 3.9.



Gambar 3.9 Pengaturan Parameter Untuk *Generate Association Rule*

Pada Gambar 3.9 ditunjukkan beberapa parameter yang harus ditetapkan sebelum melakukan *generate association rule*. Parameter *support*, *confidence* dan *lift* merupakan nilai minimum yang menjadi tolak ukur apakah sebuah *rule* dapat dikatakan *rule* yang kuat atau tidak. Setiap *rule* yang dihasilkan akan memiliki tiga nilai yang mengikutinya, yaitu nilai *lift*, *support* dan *confidence*. Oleh karena itu setiap nilai ini harus lebih besar dari nilai minimumnya agar *rule* dapat dikatakan sebagai *rule* yang kuat.

Keluaran dari apriori adalah sekumpulan *rule* yang memenuhi nilai minimum *support*, minimum *confidence* dan minimum *lift ratio*. Format keluaran apriori dengan menggunakan TANAGRA adalah Antecedent, Consequent, Lift, *Support* (%) dan *Confidence* (%). Ada tiga nilai yang harus dicari pada setiap *rule* yaitu nilai *support*, *confidence* dan *lift ratio*.

Apriori menerapkan setiap perhitungan *support*, *confidence* dan *lift ratio* pada setiap *rule*. Pada apriori scan dilakukan pada *dataset* untuk mengetahui jumlah kemunculan satu *item* dalam semua transaksi yang ada dalam *dataset*. Sebagai contoh, *rule* yang digunakan adalah *rule* pertama pada Gambar 3.10.

N	Antecedent	Consequent	Lift	Support (%)	Confidence (%)
1	"22916=true"	"22917=true"	85.30246	1.006	93.22
2	"22917=true"	"22916=true"	85.30246	1.006	92.05
3	"23170=true" - "23171=true"	"23172=true"	59.40507	1.125	80.13
4	"23172=true"	"23170=true" - "23171=true"	59.40507	1.125	83.39
5	"23170=true" - "23172=true"	"23171=true"	56.22931	1.125	94.615

Gambar 3.10 Contoh *Rule* Apriori

Pada Gambar 3.10, contoh *rule* yang diberikan adalah *rule* dengan 1-*itemset* pada bagian Antecedent dan Consequent. Untuk menghitung *support*, perlu diketahui jumlah kemunculan *item* pada Antecedent dan Consequent pada semua transaksi. Jumlah kemunculan *item* dengan kode 22916 dan 22917 secara bersama di dalam satu transaksi adalah sebanyak 220 kali. Jumlah seluruh transaksi adalah 21870. Maka perhitungan untuk mencari nilai *support* dengan menggunakan Persamaan 1 adalah :

$$\begin{aligned}
 \text{Support}(22916, 22917) &= \frac{\text{Jumlah transaksi mengandung 22916 dan 22917}}{\text{Total Transaksi}} \times 100 \\
 &= \frac{220}{21870} \times 100
 \end{aligned}$$

$$= 1,006 \%$$

Kemudian untuk mencari nilai *confidence* nya, perlu diketahui jumlah kemunculan Antecedent dan Consequent bersama dalam satu transaksi, dan jumlah kemunculan Antecedent dari semua transaksi. Kemunculan *item* 22916 dan 22917 secara bersama adalah 220 kali sedangkan kemunculan *item* 22916 adalah sebanyak 236 kali. Maka perhitungan untuk mencari nilai *confidence* dari *rule* ini dengan menggunakan Persamaan 3 adalah :

$$\begin{aligned} Conf(22916, 22917) &= \frac{\text{Jumlah transaksi mengandung 22916 dan 22917}}{\text{Jumlah transaksi mengandung 22916}} \times 100 \\ &= \frac{220}{236} \times 100 \\ &= 93,22 \% \end{aligned}$$

Untuk mencari nilai *lift ratio*, pertama kali nilai *expected confidence* harus ditentukan. Perhitungan untuk mencari *expected confidence* menggunakan Persamaan 4 adalah :

$$\begin{aligned} Expected\ confidence &= \frac{\text{Jumlah transaksi mengandung 22917}}{\text{Total Transaksi}} \\ &= \frac{239}{21870} \\ &= 0,0109282121627801 \end{aligned}$$

Kemudian nilai *lift ratio* dapat ditentukan dengan menggunakan Persamaan 5 :

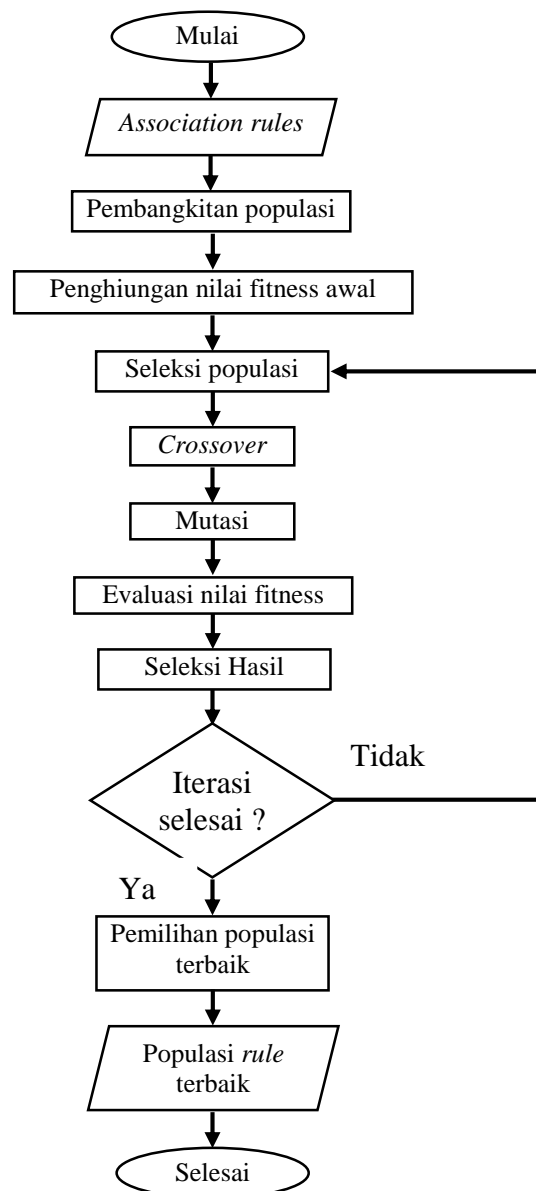
$$\begin{aligned} Lift\ ratio &= \frac{Confidence}{Expected\ Confidence} \text{ (nilai } confidence \text{ sebelum persentase)} \\ &= \frac{0.9322033898305805}{0.0109282121627801} \\ &= 85,30246 \end{aligned}$$

1.7. Evaluasi *Rule Apriori* Menggunakan Algoritma Genetika

Pada penelitian ini, algoritma genetika diterapkan pada bahasa pemrograman Python 3.6. Pemilihan bahasa pemrograman ini dilakukan karena Python merupakan bahasa pemrograman tingkat tinggi yang dapat dan mudah diaplikasikan ke dalam berbagai macam kelas masalah [24]. Selain itu, Python merupakan bahasa interpreter, interaktif, dan bahasa yang berbasis

objek yang memiliki modul yang dapat diakses dengan mudah, penulisan yang dinamis, tipe data yang dinamis, dan berbagai *class*.

Pada penelitian ini, algoritma genetika digunakan untuk mencari populasi *association rule* yang memenuhi kriteria terbaik. Tahapan algoritma genetika ditunjukkan pada Gambar 3.11.



Gambar 3.11 *Flowchart* Optimasi Rule Menggunakan Algoritma Genetika

Algoritma genetika menerapkan perhitungan fungsi objektif dan nilai *fitness* di setiap generasinya. Nilai fungsi objektif menentukan tingkat kekuatan sebuah *rule*, sedangkan nilai *fitness* digunakan pada proses seleksi populasi. Sebagai contoh *rule* yang digunakan ditunjukkan pada Gambar 3.12.

N	Antecedent	Consequent	Lift	Support (%)	Confidence (%)
1	"22916=true"	"22917=true"	85.30246	1.006	93.22
2	"22917=true"	"22916=true"	85.30246	1.006	92.05
3	"23170=true" - "23171=true"	"23172=true"	59.40507	1.125	80.13
4	"23172=true"	"23170=true" - "23171=true"	59.40507	1.125	83.39
5	"23170=true" - "23172=true"	"23171=true"	56.22931	1.125	94.615

Gambar 3.12 Contoh *Rule* Pada Populasi GA

1.1.1. Pembangkitan Populasi

Tahap awal yang dilakukan adalah pembangkitan awal populasi sebanyak jumlah kromosom. Pada penelitian ini, populasi awal dibangkitkan dengan memilih Antecedent dan Consequent secara acak yang diambil dari kumpulan *rule* yang dihasilkan oleh apriori, yaitu sebanyak 958 *rule*. Format *rule* pada setiap kromosom memiliki lima elemen, yaitu Antecedent, Consequent, *lift*, *support* dan *confidence*. Format kromosom dalam populasi yang digunakan pada algoritma genetika hanya memiliki tiga elemen, yaitu Antecedent, Consequent dan fungsi objektif.

Inisialisasi populasi adalah tahap awal yang harus dilakukan pada algoritma genetika untuk menentukan populasi pertama yang akan diproses sejumlah iterasi yang ditentukan. Dengan mengambil contoh pada Gambar 3.12, maka populasi awal adalah :

Kromosom[1] = ["22916=true" ; "22917=true"]

Kromosom[2] = ["22917=true" ; "22916=true"]

Kromosom[3] = ["23170=true" - "23171=true" ; "23172=true"]

Kromosom[4] = ["23172=true" ; "23170=true" - "23171=true"]

Kromosom[5] = ["23170=true" ; "23171=true"]

1.1.2. Penghitungan Awal Nilai Fungsi Objektif

Nilai fungsi objektif adalah fungsi yang menyatakan baik tidaknya suatu solusi dan dijadikan sebagai acuan dalam mencapai nilai optimal terhadap masalah yang diteliti. Dalam penelitian ini, nilai fungsi objektif dihitung dengan menggunakan nilai *support* dan *confidence* sebagai parameter utama. Nilai fungsi objektif untuk setiap kromosom (*association rule*) dapat dihitung dengan menggunakan Persamaan 7.

Sebagai contoh untuk menghitung fungsi objektif semua *rule* pada Gambar 3.12, nilai yang dilibatkan adalah nilai *support* dan nilai *confidence*. Sebagai contoh perhitungan nilai fungsi objektif pada *rule* pertama adalah :

$$\begin{aligned}
 f(1) &= (a \times \text{supp}(1)) + (b \times \text{conf}(1)) \\
 &= (0,5 \times 0,01006) + (0,5 \times 0,9322) \\
 &= 0,00503 + 0,4661 \\
 &= 0,47113
 \end{aligned}$$

1.1.3. Seleksi Populasi

Proses seleksi populasi dilakukan dengan menggunakan metode *roulette wheel selection* dimana setiap setiap kromosom memiliki nilai batas distribusi probabilitas yang menunjukkan kemungkinan kromosom ini akan terpilih pada populasi selanjutnya. Proses seleksi dilakukan agar kromosom yang mempunyai nilai *fitness* yang kecil mempunyai probabilitas yang lebih besar untuk terpilih pada populasi selanjutnya. Ada beberapa langkah yang harus dilakukan untuk proses seleksi, yaitu :

a. mencari nilai *fitness*

Nilai *fitness* digunakan sebagai parameter perhitungan pada proses seleksi kromosom untuk populasi selanjutnya. Nilai *fitness* dipengaruhi oleh besarnya fungsi objektif yang dimiliki oleh sebuah kromosom, oleh karena itu setelah fungsi objektif untuk setiap kromosom telah didapatkan, nilai *fitness* dapat dihitung dengan menggunakan Persamaan 8. Sebagai contoh, untuk menghitung nilai *fitness rule* pertama pada Gambar 3.12 adalah :

$$\begin{aligned}
 \text{fit}(1) &= \frac{1}{1+f(1)} \\
 &= \frac{1}{1+0,47113} \\
 &= 0,6797
 \end{aligned}$$

Proses seleksi menggunakan parameter nilai *fitness* untuk menentukan kromosom apa saja yang dapat dipilih untuk populasi selanjutnya. Dengan menggunakan contoh *rule* pada Gambar 3.12, langkah pertama adalah menentukan semua nilai *fitness* setiap kromosom. Nilai *fitness* ditentukan dengan menggunakan Persamaan 8, sehingga hasilnya adalah :

$$\text{fitness}[1] = 0,47113$$

$$\text{fitness}[2] = 0,68246$$

$$\text{fitness}[3] = 0,71109$$

$$\text{fitness}[4] = 0,70295$$

$$\text{fitness}[5] = 0,6773$$

b. mencari probabilitas

Nilai probabilitas setiap kromosom dihitung dengan menggunakan Persamaan 9, dimana setiap nilai *fitness* dibagi dengan jumlah total nilai *fitness* yang didapatkan. Perhitungannya adalah sebagai berikut :

$$\begin{aligned}
 \text{Total fitness} &= 0,47113 + 0,68246 + 0,71109 + 0,70295 + 0,6773 \\
 &= 3,24493
 \end{aligned}$$

$$P[1] = \frac{\text{fitness}[1]}{\text{total fitness}} = \frac{0,47113}{3,24493} = 0,14519$$

$$P[2] = \frac{\text{fitness}[2]}{\text{total fitness}} = \frac{0,68246}{3,24493} = 0,21032$$

$$P[3] = \frac{\text{fitness}[3]}{\text{total fitness}} = \frac{0,71109}{3,24493} = 0,21914$$

$$P[4] = \frac{\text{fitness}[4]}{\text{total fitness}} = \frac{0,70295}{3,24493} = 0,21663$$

$$P[5] = \frac{\text{fitness}[5]}{\text{total fitness}} = \frac{0,6773}{3,24493} = 0,20872$$

c. mencari komulatif probabilitas

Setelah nilai probabilitas didapatkan, langkah selanjutnya adalah mencari nilai komulatif probabilitas dari setiap kromosom menggunakan Persamaan 10. Perhitungannya adalah sebagai berikut :

$$C[1] = \sum_{k=1}^1 P[1] = 0,14519$$

$$C[2] = \sum_{k=1}^2 P[2] = 0,14519 + 0,21032 = 0,35551$$

$$C[3] = \sum_{k=1}^3 P[3] = 0,14519 + 0,21032 + 0,21914 = 0,57465$$

$$C[4] = \sum_{k=1}^4 P[4] = 0,14519 + 0,21032 + 0,21914 + 0,21663 = 0,79128$$

$$C[5] = \sum_{k=1}^5 P[5] = 0,14519 + 0,21032 + 0,21914 + 0,21663 + 0,20872 \\ = 1$$

d. memilih populasi selanjutnya

Setelah nilai komulatif probabilitas diketahui, seleksi menggunakan *roulette wheel* dapat dilakukan. Selanjutnya bilangan acak dibangkitkan sesuai dengan jumlah kromosom dalam rentang 0 sampai 1.

$$R[1] = 0,20183$$

$$R[2] = 0,29814$$

$$R[3] = 0,05019$$

$$R[4] = 0,81232$$

$$R[5] = 0,56041$$

Proses seleksi adalah dengan membandingkan setiap angka acak yang dibangkitkan dengan nilai komulatif probabilitasnya. Jika $R[x] < C[1]$, maka pilih kromosom 1 sebagai induk. Sebagai contoh yang lain, pilih kromosom ke-x sebagai induk jika $C[x-1] < R[k] < C[x]$. Seleksi akan dilakukan sebanyak jumlah kromosom. Dari contoh bilangan acak di atas, maka populasi kromosom baru yang didapatkan adalah :

$$\text{Kromosom}[1] = \text{Kromosom}[2] = ["22917=true" ; "22916=true"]$$

$$\text{Kromosom}[2] = \text{Kromosom}[2] = ["22917=true" ; "22916=true"]$$

Kromosom[3] = Kromosom[1] = ["22916=true" ; "22917=true"]
 Kromosom[4] = Kromosom[5] = ["23170=true" ; "23171=true"]
 Kromosom[5] = Kromosom[3] = ["23170=true" – "23171=true" ;
 = "23172=true"]

1.1.4. *Crossover*

Crossover adalah proses kawin silang dan tukar gen dengan memilih satu atau lebih posisi acak dalam sebuah kromosom kemudian sejumlah gen ditukarkan posisinya dengan posisi gen pada kromosom yang lain. Untuk melakukan *crossover* dibutuhkan dua buah *parent* yang dipilih secara acak. Jumlah *parent* yang mengalami *crossover* dipengaruhi oleh parameter *crossover rate*.

Teknik *crossover* yang akan digunakan dalam penelitian ini adalah teknik *one-point crossover* dimana dipilih satu posisi tetap pada kromosom, kemudian gen yang dipotong oleh satu posisi ini saling ditukarkan diantara kedua *parent*. Teknik *one-point crossover* juga digunakan karena format kromosom yang digunakan pada penelitian ini hanya memiliki dua elemen, yaitu Antecedent dan Consequent. Kedua elemen ini yang akan digunakan untuk *crossover*. Langkah - langkah *crossover* adalah :

- a. menentukan *crossover rate* dan membangkitkan bilangan acak

Crossover rate merupakan nilai yang mempengaruhi jumlah kromosom yang mengalami proses *crossover*/kawin silang. Misalkan *crossover rate* adalah 50%, maka diharapkan ada setengah kromosom yang berubah karena mengalami *crossover*.

Metode *crossover* yang digunakan adalah *one point crossover* karena format kromosom hanya mengandung dua parameter yaitu Antecedent dan Consequent *rule*. Proses pertama adalah menentukan parameter *crossover rate* dan membangkitkan bilangan acak sesuai dengan jumlah kromosom. Dengan melanjutkan dari proses sebelumnya, maka perhitungannya adalah :

$$Crossover\ rate = 50\% = 0,5$$

$$R[1] = 0,19$$

$$R[2] = 0,76$$

$$R[3] = 0,81$$

$$R[4] = 0,25$$

$$R[5] = 0,34$$

b. menentukan kromosom yang mengalami *crossover*

Pada langkah ini bilangan acak dibangkitkan sebanyak jumlah kromosom. Jika bilangan acak ini ke-x ($R[x]$) lebih kecil dari *crossover rate*, maka ambil kromosom ke-x ($\text{chromosome}[x]$) sebagai *parent*.

c. mencari *offspring*

Offspring adalah anak yang dihasilkan dari hasil *crossover* dari dua buah *parent*. *Parent* yang digunakan adalah format kromosom dengan dua elemen, yaitu Antecedent dan Consequent. Dengan menggunakan teknik *one-point crossover*, nilai Antecedent dari kedua *parent* ditukarkan sehingga menghasilkan sebuah *offspring*. Sebagai gambaran, setiap *offspring* ke-x ($\text{offspring}[x]$) ditentukan dengan menukar nilai Antecedent pada $\text{parent}[x]$ dan nilai Antecedent pada $\text{parent}[x+1]$. Setelah seluruh *offspring* didapatkan sesuai dengan jumlah *parent*, selanjutnya seluruh *offspring* akan menggantikan posisi kromosom *parent* sebelumnya.

Pemilihan induk adalah dengan membandingkan nilai bilangan acak dengan *crossover rate* yang ditentukan. Kromosom ke-x akan dijadikan induk jika $R[k] < \text{crossover rate}$. Dari contoh di atas, maka kromosom yang dijadikan induk adalah kromosom[1], kromosom[4], dan kromosom[5]. Posisi *cut-point crossover* adalah pada posisi 1, yaitu antara gen pertama dan kedua. Setelah induk diketahui, maka proses *crossover* dapat dilakukan dengan menukar gen yang ada diantara dua kromosom yang menjadi induk. Nilai yang menjadi hasil *crossover* adalah *offspring* yang merupakan hasil kawin silang antara dua induk.

$$\begin{aligned} \text{Offspring}[1] &= \text{kromosom}[1] \times \text{kromosom}[4] \\ &= [“22917=true” ; “22916=true”] \times [“23170=true” \\ &\quad ; “23171=true”] \end{aligned}$$

$$\begin{aligned}
&= ["22917=true"; "23171=true"] \\
\text{Offspring}[4] &= \text{kromosom}[4] \times \text{kromosom}[5] \\
&= ["23170=true"; "23171=true"] \times ["23170=true" \\
&\quad - "23171=true"; "23172=true"] \\
&= ["23170=true"; "23172=true"] \\
\text{Offspring}[5] &= \text{kromosom}[5] \times \text{kromosom}[1] \\
&= ["23170=true" - "23171=true"; "23172=true"] \\
&\quad \times ["22917=true"; "22916=true"] \\
&= ["23170=true" - "23171=true"; "22916=true"]
\end{aligned}$$

Offspring yang telah didapatkan akan menggantikan kromosom *parent* pada posisinya. Dengan demikian populasi kromosom setelah mengalami proses *crossover* menjadi :

$$\text{Kromosom}[1] = ["22917=true"; "23171=true"]$$

$$\text{Kromosom}[2] = ["22917=true"; "22916=true"]$$

$$\text{Kromosom}[3] = ["22916=true"; "22917=true"]$$

$$\text{Kromosom}[4] = ["23170=true"; "23172=true"]$$

$$\text{Kromosom}[5] = ["23170=true" - "23171=true"; "22916=true"]$$

1.1.5. Mutasi

Mutasi dalam algoritma genetika adalah proses mengganti satu gen dalam sebuah kromosom menjadi satu nilai baru yang bersifat acak (dalam biner, nilainya di-*inverse*). Karena bentuk kromosom yang digunakan hanya memiliki nilai Antecedent dan Consequent, maka gen yang akan diganti akan berupa sebuah *itemset* karena Antecedent dan Consequent sendiri hanya mengandung satu atau lebih *itemset*. Langkah - langkah mutasi adalah :

- a. menentukan *mutation rate*, panjang total gen dan jumlah mutasi

Mutation rate adalah nilai yang menentukan berapa banyak proses mutasi yang harus dilakukan pada populasi kromosom. Panjang total gen didapatkan dari perkalian antara jumlah gen dalam kromosom dan

jumlah populasi. Pada penelitian ini, jumlah gen dalam satu kromosom hanya ada dua, yaitu Antecedent dan Consequent.

Proses mutasi dalam satu populasi ditentukan oleh *mutation rate*. Mutasi dimulai dengan penentuan panjang total gen dan *mutation rate*. Jumlah mutasi yang terjadi adalah *mutation rate* dikali dengan panjang gen. Perhitungannya adalah :

$$\begin{aligned} \text{Panjang total gen} &= (\text{jumlah gen dalam kromosom}) * \text{jumlah kromosom} \\ &= 2 * 5 \\ &= 10 \\ \text{Mutation rate} &= 20\% = 0,2 \\ \text{Jumlah mutasi} &= 0,2 * 10 \\ &= 2 \end{aligned}$$

b. membangkitkan bilangan acak dan proses mutasi

Setelah jumlah mutasi ditentukan, selanjutnya bilangan acak akan dibangkitkan sebanyak jumlah mutasi dengan range dari 1 sampai 10 (panjang total gen). Misalkan bilangan acak yang dibangkitkan adalah ($R = [3, 8]$). Posisi gen yang akan mengalami mutasi adalah kromosom ke-2 gen 1 dan kromosom ke-4 gen 2. Nilai gen akan mengalami mutasi sesuai dengan nilai yang terdapat pada populasi besar *rule* yang dihasilkan oleh apriori.

$$\begin{aligned} \text{Kromosom}[2] &= ["22917=true" ; "22916=true"] \\ &= ["\mathbf{23174=true}" ; "22916=true"] \\ \text{Kromosom}[4] &= ["23170=true" ; "23172=true"] \\ &= ["23170=true" ; "\mathbf{23175=true}"] \end{aligned}$$

Setelah mengalami mutasi, maka populasi selanjutnya adalah :

$$\begin{aligned} \text{Kromosom}[1] &= ["22917=true" ; "23171=true"] \\ \text{Kromosom}[2] &= ["23174=true" ; "22916=true"] \\ \text{Kromosom}[3] &= ["22916=true" ; "22917=true"] \\ \text{Kromosom}[4] &= ["23170=true" ; "23175=true"] \\ \text{Kromosom}[5] &= ["23170=true" - "23171=true" ; "22916=true"] \end{aligned}$$

1.1.6. Evaluasi Nilai Fungsi Objektif

Proses ini menghitung kembali nilai fungsi objektif semua kromosom yang ada di dalam populasi setelah melalui proses *crossover* dan mutasi. Fungsi objektif dalam satu populasi akan berbeda dengan fungsi objektif pada populasi sebelumnya karena proses *crossover* dan mutasi akan mengubah beberapa elemen kromosom yang ada di dalam populasi.

1.1.7. Seleksi Kromosom

Seleksi kromosom dilakukan untuk menyeleksi semua kromosom / *rule* yang valid dengan syarat nilai fungsi objektif harus lebih besar dari *minimum confidence* yang telah ditentukan. Selain kromosom / *rule* yang valid, parameter lain yang dicari adalah nilai rata-rata fungsi objektif yang didapatkan dari total jumlah fungsi objektif dari *rule* yang valid dibagi dengan jumlah *rule* yang valid.

Jumlah kromosom yang valid dan nilai rata-rata fungsi objektif akan digunakan sebagai parameter untuk mengevaluasi iterasi dengan populasi yang terbaik.

1.1.8. Ekstraksi Kromosom

Ekstraksi kromosom dilakukan untuk mengambil setiap kromosom yang valid dimana nilai fungsi objektifnya lebih besar dari nilai *minimum confidence*. Kromosom yang valid pada setiap iterasi akan dikumpulkan pada satu populasi khusus sebagai populasi *rule* valid yang telah ditemukan oleh algoritma genetika. Populasi ini yang kemudian akan menjadi hasil keluaran dari algoritma genetika setelah melakukan proses genetik sebanyak jumlah iterasi yang telah ditentukan.