BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terkait

2.1.1. Penelitian M. Dhanabhakyam and M. Punithavalli [3]

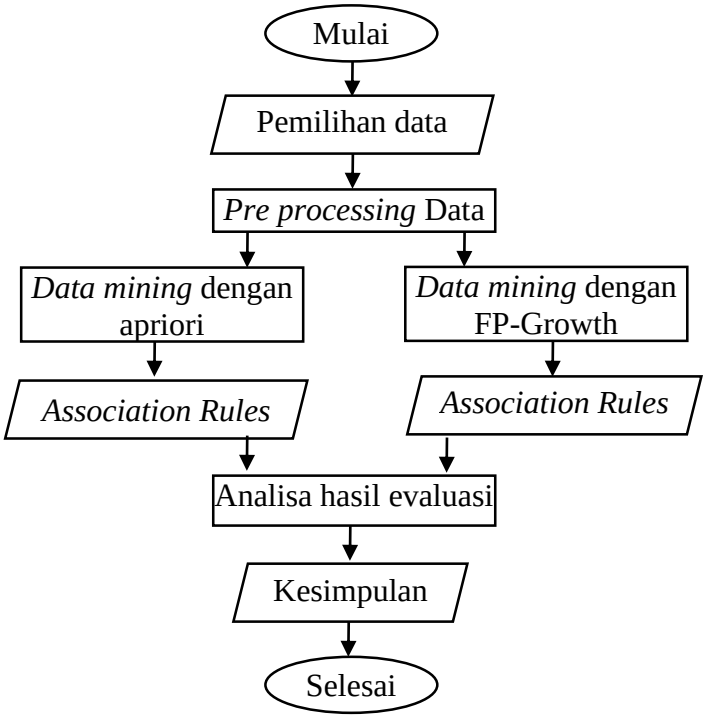
Penelitian ini melakukan survey algoritma *data mining* untuk *market basket analysis.* Penelitian ini membandingkan enam belas penelitian sebelumnya yang membahas tentang algoritma yang digunakan dalam proses *association rule mining.* Di antara algoritma yang digunakan untuk *data mining,* algoritma apriori ditemukan lebih baik diterapkan dalam *market basket analysis.* Meskipun begitu, masih ada beberapa kelemahan yang dimiliki oleh algoritma apriori. Kelemahan ini antara lain :

a. Algoritma apriori melakukan banyak kali *scan* terhadap *dataset.* Setiap kali ada tambahan pilihan yang dibuat selama proses *scan* akan menciptakan pekerjaan tambahan untuk *database* yang digunakan, sehingga *database* harus menyimpan sejumlah besar layanan data. Hal ini menyebabkan kurangnya kapasitas memori untuk menyimpan data tambahan. Hasil ini mempunyai efisiensi yang rendah.
b. *Frequent itemset* dalam keadaan *dataset* yang lebih besar menyebabkan peningkatan yang signifikan dalam waktu komputasi.
c. Penyempitan hasil algoritma. Dalam hal ini algoritma tidak membuat hasil yang lebih baik. Oleh karena itu diperlukan suatu pengembangan atau modifikasi pada algoritma apriori.

Kelemahan ini dapat diatasi dengan memodifikasi algoritma apriori secara efektif. Algoritma apriori memiliki kemungkinan untuk menyebabkan kurangnya akurasi dalam menentukan peraturan asosiasi terutama jika diterapkan pada sebuah *dataset* dengan jumlah data yang besar. Untuk mengatasinya, algoritma lain dapat dikombinasikan dengan algoritma apriori. Ini akan membantu dalam pemilihan aturan asosiasi yang lebih baik khususnya untuk *market basket analysis*.

2.1.2. **Penelitian G. Gunadi dan D. I. Sensuse** [9]

Penelitian ini melakukan perbandingan algoritma apriori dan algoritma *frequent pattern growth* (FP-growth) untuk penentuan algoritma yang paling baik dan sesuai (*best-fit algorithm*) untuk permasalahan yang diteliti pada percetakan PT. Gramedia. Langkah penelitian ini ditunjukkan pada Gambar 2.1.



Gambar 2.1 *Flowchart* Aktivitas Penelitian

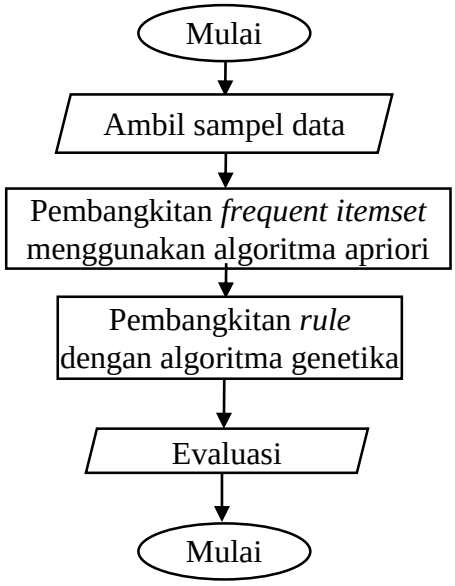
Dalam pemilihan data, yang digunakan dalam penelitian ini adalah data transaksi pesanan di Percetakan PT. Gramedia yang terkumpul mulai dari tanggal 1 Januari 2005 hingga 1 Agustus 2010. Data ini kemudian diproses menjadi bentuk *dataset* yang diseleksi berdasarkan jumlah *frequent itemset* secara keseluruhan dengan nilai minimum sebesar 0,2 %. Setelah data terpilih, dilakukan *pre-processing* terhadap *dataset* yang telah terbentuk sehingga menjadi tiga buah *dataset* dalam bentuk tabel dengan nama Tbl_Dataset_Training, Tbl_Dataset_Evaluation1 dan Tbl_Dataset_Evaluation2. Tbl_Dataset_Training diisi dengan seluruh data transaksi penjualan yang terdapat pada basis data penelitian, Tbl_Dataset_Evaluation1 diisi dengan 5.000 data transaksi penjualan

terakhir yang terdapat pada basis data penelitian, dan Tbl_Dataset_Evaluation2 diisi dengan 10.000 data transaksi penjualan yang diambil secara acak dari basis data penelitian.

Proses *association rule mining* menggunakan algoritma apriori dan FP-Growth dengan keluaran sekumpulan *association rule*. Pada tahap evaluasi, ada dua faktor yang dievaluasi dalam penggunaan kedua algoritma yang digunakan, yaitu ukuran generalitas dan ukuran reliabilitas dari aturan asosiasi yang dihasilkan. Dalam ukuran generalitas terdapat nilai *support* dan nilai *coverage*, sementara pada ukuran realibilitas memiliki nilai *confidence, added value* dan *correlation.* Hasil dari penelitian ini menunjukkan bahwa apriori memiliki tingkat akurasi yang lebih tinggi dibandingkan FP-growth dengan persentase 257,4543 % lebih tinggi.

2.1.3. Penelitian Shanta Rangaswamy dan Ahobha G. [8]

Penelitian ini melakukan proses optimasi *association rule mining* menggunakan algoritma genetika. Langkah yang ada dalam penelitian ini ditunjukkan pada Gambar 2.2



Gambar 2.2 *Flowchart* Aktivitas Penelitian 2

Data yang digunakan dalam penelitian ini adalah data *log* web tentang interaksi *user* ketika melakukan *request resource* pada web yang berbeda.

Menganalisa data ini dapat membantu memahami *user behavior* dan struktur web sehingga dapat digunakan untuk memperbaiki desain dari *resource* yang ada. Selanjutnya dalam proses *association rule mining* menggunakan OARM (*Optimized Association Rule Mining*. Proses ini terbagi menjadi dua bagian, pembangkitan *frequent itemset* menggunakan algoritma apriori dan pembangkitan *rule* menggunakan algoritma genetika. Dengan menggunakan algoritma genetika, sistem dapat memprediksi *rule* yang mengandung atribut negatif dalam proses pengambilan *rule* yang memiliki lebih dari satu atribut.

Hasil penelitian ini adalah *rule* yang mengandung atribut positif. *Rule* yang didapatkan merupakan *rule* yang hasil optimasi menggunakan algoritma genetika yang diaplikasikan pada *frequent itemset* hasil *association rule mining* menggunakan algoritma apriori.

2.1.4. **Penelitian Mohit K. Gupta dan Geeta Sikka** [5]

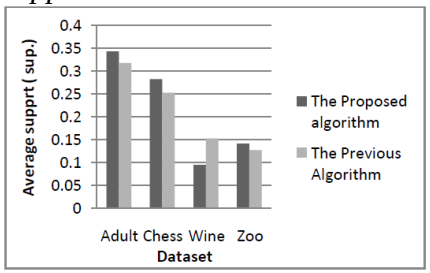
Penelitian ini melakukan proses *association rule mining* menggunakan *Multi-objectie Feature* pada algoritma genetika. Untuk evauasi *rule* yang dihasilkan menggunakan parameter *support, confidence, comprehensibility* dan *interestingness.* Objek penelitian yang digunakan adalah empat *dataset* yang diambil dari UCI Machine Learning Repository yaitu *dataset* Adult, Chess, Wine dan Zoo. Untuk setiap pengujian pada *dataset*, parameter algoritma genetika yang digunakan adalah 100 individu dalam satu populasi dan dieksekusi sebanyak 200 iterasi. Eksperimen menggunakan algoritma genetika akan dievaluasi dan dibandingkan dengan hasil dari apriori dan hasil dari algoritma genetika yang dilakukan oleh M. Ramesh, dkk [6]. Hasil pengujian yang didapatkan adalah hasil rata-rata setelah 10 eksekusi.

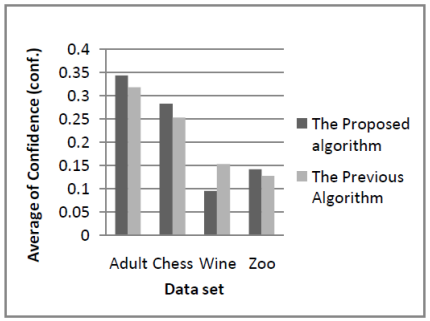
Kesimpulan yang didapatkan adalah *association rule* yang dihasilkan dari algoritma yang diajukan dapat dikatakan mengalami peningkatan performa dalam hal ekstraksi *association rule* yang menarik. Jumlah *rule* yang didapatkan lebih sedikit jika dibandingkan dengan apriori dan teknik

sebelumnya [6]. Hal ini menandakan bahwa optimasi *association rule* yang dilakukan bersifat efisien dan efektif.

Tabel 2.1 Daftar Penelitian Terkait

No.	Nama penulis	Judul Penelitian	Tahun	Masalah	Hasil
1.	Dr. M. Dhanabhakyam dan M. Punithavalli	A Survey on Data Mining Algorithm for Market Basket Analysis	2011	Belum ada perbandingan algoritma terbaik dalam metode *market basket analysis* secara umum.	Berdasarkan perbandingan enam belas penelitian sebelumnya tentang algoritma dalam *association rule mining*, penelitian ini menyimpulkan bahwa algoritma apriori merupakan algoritma yang lebih baik diterapkan.
2.	Goldie Gunadi dan Dana Indra Sensuse	Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori Dan Frequent Pattern Growth (FP-Growth) : Studi Kasus Percetakan PT. Gramedia	2012	Percetakan PT. Gramedia memiliki data transaksi pesanan percetakan yang besar namun belum mampu membuat strategi pemasaran dan panjualan yang efektif.	Dari hasil analisa diketahui bahwa tingkat kekuatan aturan asosiasi yang dihasilkan apriori lebih besar dibandingkan FP-Growth karena nilai *support* yang dihasilkan apriori lebih tinggi. Algoritma apriori unggul dibanding FP-Growth pada tingkat akurasi 257,4543 %.
3.	Shanta Rangaswamy dan Shobha G.	Optimized Association Rule Mining Using Genetic Algorithm	2009	Dalam mengakses informasi dari *log file* yang besar dalam *database* pada sisi server dapat memakan waktu yang cukup lama dan tidak efisien. *Association rule mining* diterapkan dalam proses ini, namun *rule* yang dihasilkan oleh algoritma apriori belum dapat membedakan *rule* dengan atribut positif dan negatif.	Hasil penelitian ini adalah *rule* yang mengandung atribut positif. *Rule* yang didapatkan merupakan *rule* yang hasil optimasi menggunakan algoritma genetika yang diaplikasikan pada *frequent itemset* hasil *association rule mining* menggunakan algoritma apriori.

<table>
<tr>
<td>4</td>
<td>Mohit K. Gupta dan Geeta Sikka</td>
<td>Association Rule Extraction using Multi-objective Feature of Genetic Algorithm</td>
<td>2013</td>
<td>Proses association rule mining pada umumnya menghasilkan rule dalam jumlah yang sangat besar sehingga membuat pekerjaan seorang database analyst bertambah untuk menyeleksi lagi rule yang menarik. Selain itu, algoritma sebelumnya hanya menggunakan satu parameter (min_supp) untuk mendapatkan rule yang menarik.</td>
<td>Perbandingan berdasarkan support dan confidence :
<table>
<tr><th>Dataset</th><th>Algorithms</th><th>Sup</th><th>Conf</th></tr>
<tr><td rowspan="2">Adult</td><td>The Proposed Algorithm</td><td>.343</td><td>1.000</td></tr>
<tr><td>The Previous Algorithm</td><td>.318</td><td>1.000</td></tr>
<tr><td rowspan="2">Chess</td><td>The Proposed Algorithm</td><td>.283</td><td>.920</td></tr>
<tr><td>The Previous Algorithm</td><td>.253</td><td>.908</td></tr>
<tr><td rowspan="2">Wine</td><td>The Proposed Algorithm</td><td>.095</td><td>.740</td></tr>
<tr><td>The Previous Algorithm</td><td>.153</td><td>.680</td></tr>
<tr><td rowspan="2">Zoo</td><td>The Proposed Algorithm</td><td>.142</td><td>.603</td></tr>
<tr><td>The Previous Algorithm</td><td>.127</td><td>.540</td></tr>
</table>
Perbandingan berdasarkan rata-rata support :


Perbandingan berdasarkan rata-rata confidence :


</td>
</tr>
</table>

2.2. Dasar Teori

2.2.1. Data Mining

Secara sederhana, *data mining* mengacu pada pengambilan/penggalian pengetahuan dari gudang basis data yang besar. Menurut Turban dkk (2005) [10] dalam bukunya menjelaskan bahwa *data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data yang besar. Banyak orang yang menganggap *data mining* sebagai sinonim dari istilah lainnya, yaitu *Knowledge Discovery from Data* atau KDD. *Knowledge discovery* sebagai sebuah proses terdiri dari tujuh langkah pelaksanaan, yaitu d*ata cleaning, data integration, data selection, data transformation, data mining, pattern evaluation,* dan k*nowledge presentation* [11]. Langkah pertama sampai keempat merupakan bentuk berbeda dari *preprocessing* data, dimana data dipersiapkan untuk kegiatan *mining. Data mining* berinteraksi langsung dengan pengguna atau sebuah basis pengetahuan. Pola yang unik dapat diberikan kepada pengguna dan dapat disimpan sebagai pengetahuan yang baru. Dalam proses KDD, *data mining* hanya salah satu langkah dalam seluruh proses yang ada, namun merupakan yang paling penting karena *data mining* mengungkapkan sebuah pola tersembunyi dan pengetahuan baru untuk sebuah evaluasi.

Dalam pandangan yang lebih luas tentang fungsionalitasnya, *data mining* adalah sebuah proses pencarian pengetahuan menarik dari kumpulan data yang tersimpan di dalam *database, data warehouse* atau penyimpanan informasi yang lainnya. Berdasarkan pandangan ini, arsitektur *data mining* memiliki beberapa komponen utama [10], yaitu :

a. *Database, data warehouse, World Wide Web*. Merupakan satu atau beberapa *database, data warehouse, spreadsheet* atau jenis repositori informasi lainnya. *Data cleaning* dan *data integration* dapat dilakukan pada komponen ini.

b. *Server database/data warehouse. Server* ini bertanggung jawab untuk pengambilan data yang relevan berdasarkan pada permintaan data oleh pengguna.
c. *Knowledge base.* Merupakan daerah pengetahuan yang digunakan untuk mengarahkan pencarian atau mengevaluasi ketertarikan dalam menghasilkan sebuah pola. Contoh dari daerah pengetahuan adalah kendala atau batasan tambahan dan *metadata* (seperti penggambaran data dari sumber yang berbeda).
d. *Data mining engine.* Hal ini adalah komponen utama *data mining* yang terdiri dari satu set modul fungsional untuk tugas seperti karakterisasi, analisis korelasi dan asosiasi, klasifikasi, prediksi, analisis *cluster,* analisis *outlier* dan analisis evolusi.
e. *Pattern Evaluation Module.* Komponen ini berinteraksi dengan modul *data mining* agar pencarian terfokus pada pola yang menarik. Modul ini dapat diintegrasikan dengan modul *data mining,* tergantung pada metode *data mining* yang digunakan
f. *User Interface.* Komponen ini berkomunikasi diantara pengguna dan sistem *data mining,* memungkinkan pengguna untuk berinteraksi dengan sistem dengan melakukan spesifikasi tugas atau *query data mining,* memberikan informasi untuk membantu pemfokusan pencarian dan melakukan eksplorasi *data mining* berdasarkan hasil rata-rata. Selain itu, komponen ini memungkinkan pengguna untuk mengakses skema *database* dan *data warehouse,* mengevaluasi pola yang telah ditemukan dan melakukan visualisasi pola dalam bentuk yang berbeda.

Dalam data mining memiliki lima metode yang berbeda yaitu klasifikasi, *clustering, predictive modeling,* regresi dan asosiasi.

1. Klasifikasi merupakan metode *data mining* yang digunakan untuk memprediksi *class* dan properti untuk setiap *instance* data. Klasifikasi bekerja dengan cara mempelajari pola historis dari data berupa informasi seperti ciri-ciri, variabel dan fitur pada berbagai karakteristik

item yang telah diberi label sebelumnya, kemudian menempatkan objek baru ke dalam kelompok atau kelas masing-masing.

2. *Clustering* merupakan metode *data mining* yang digunakan untuk mengelompokkan data-data dengan karakteristik yang sama pada satu kelompok yang sama. Berbeda dengan klasifikasi, *clustering* tidak mempertimbangkan aspek historis dan variabel yang ada dalam suatu data dalam pengelompokannya.
3. Estimasi merupakan metode untuk menerka sebuah nilai yang belum diketahui. Metode yang digunakan antara lain *Point Estimation* dan *Confidence Interval Estimations, Simple Linear Regression and Correlation*, dan *Multiple Regression*.
4. Prediksi merupakan proses untuk menemukan pola dari data dengan menggunakan beberapa variabel untuk memprediksi variabel lain yang tidak diketahui jenis atau nilainya.
5. Asosiasi merupakan metode yang menemukan hubungan menarik yang menghubungkan data satu dengan data yang lain dalam kurun waktu tertentu. Metode asosiasi akan membuat aturan yang menyimpulkan hubungan yang telah didapatkan.

2.2.2. Analisis Asosiasi

Association analysis atau analisis asosiasi adalah salah satu metode dalam *data mining* yang berguna untuk mencari hubungan menarik yang tersembunyi di dalam *dataset* yang besar [12]. Analisis asosiasi mempunyai bidang penerapan yang luas, seperti *market basket analysis*, diagnosa medis, analisis navigasi *website*, pendidikan, finansial dan domain bisnis. Dengan banyaknya data yang terus-menerus dikumpulkan dan disimpan di dalam *database* membuat banyak perusahaan yang tertarik untuk menemukan *association rule* di dalamnya untuk meningkatkan keuntungan mereka. *Association rule* merupakan representasi dari pola informasi yang telah didapatkan dari analisis asosiasi. Salah satu contoh analisis asosiasi adalah pencarian hubungan

asosiasi diantara sejumlah besar laporan transaksi untuk membantu mendesain katalog, pemasaran silang (*cross marketing*) dan proses pengambilan keputusan bisnis yang lain.

Analisis asosiasi dimulai dari sebuah *itemset* $I=\{I_1, I_2, ..., I_m\}$. Misalkan T merupakan sebuah simbol dari data yang relevant, menjadi satu set transaksi *database* dimana setiap transaksi T adalah sebuah kumpulan item yang dilambangkan $T \subseteq I$. Setiap transaksi diasosiasikan dengan sebuah identitas/*identifier,* yang disebut TID. Misalkan A adalah sebuah satu set item yang terdapat dalam satu transaksi. A transaksi T dapat dikatakan mengandung A jika dan hanya jika $A \subseteq T$. Sebuah *association rule* adalah implikasi yang terjadi dari A ke B ($A \Rightarrow B$), dimana B adalah satu set item yang lain. Dalam hal ini berlaku $A \subset I$, $B \subset I$ dan $A \cap B = \Phi$. Aturan $A \Rightarrow B$ mengandung set transaksi D dengan *support* ***s***, dimana **s** adalah persentase transaksi D yang mengandung $A \cup B$. Aturan $A \Rightarrow B$ juga memiliki *confidence* **c** di dalam transaksi D, dimana **c** adalah persentase transaksi di dalam D yang mengandung A dan B. Dari penjelasan di atas, nilai *support*($A \Rightarrow B$) = Prob{ $A \cup B$ } dan nilai *confidence*($A \Rightarrow B$) = Prob{B/A} [13].

Aturan yang memenuhi minimum *support treshold(minsup)* dan minimum *confidence treshold(minconf)* dapat dikatakan aturan yang kuat. *Support* merupakan nilai ukuran yang penting karena aturan yang mempunyai nilai *support* yang rendah dapat terjadi secara tidak sengaja. Aturan ini cenderung tidak menarik dari segi perspektif bisnis karena dapat dianggap tidak *profitable* untuk mempromosikan item yang konsumen beli bersama. Karena alasan ini, *support* seringkali dijadikan tolak ukur untuk mengeliminasi *uninteresting rules* atau aturan yang lemah. Nilai *support* juga dimanfaatkan untuk penemuan aturan asosiasi yang efisien.

Confidence, di sisi lain mengukur reliabilitas kesimpulan yang dibuat oleh aturan. Untuk aturan $A \Rightarrow B$, semakin tinggi nilai *confidence,* semakin tinggi kemungkinan item A hadir juga dalam transaksi yang mengandung

item B. *Confidence* juga memberikan perkiraan probabilitas bersyarat dari A yang diberikan kepada B.

Hasil analisis asosiasi harus ditafsirkan dengan hati-hati. Kesimpulan yang dibuat oleh aturan tidak selalu bersifat kausal. Sebaliknya, hal ini menunjukkan hubungan koperatif yang kuat diantara item akibat dari peraturan tersebut. Kausalitas, di sisi lain memerlukan pengetahuan tentang atribut kausal dan efek dalam data yang biasanya melibatkan hubungan yang terjadi dari waktu ke waktu.

Langkah-langkah dalam *mining* dapat dijabarkan dalam langkah-langkah berikut. Diberikan sebuah set transaksi T, dan ditemukan semua *rule* mempunyai *support* ≥ *minsup* dan *confidence* ≥ *minconf*, dimana *minsup* dan *minconf* adalah batas minimal *support* dan *confidence.* Pendekatan *brute-force* untuk *mining* aturan asosiasi adalah untuk menghitung nilai *support* dan *confidence* untuk setiap *rule* yang mungkin terjadi. Pendekatan ini cukup mahal karena ada banyak *rule* yang bisa diekstraksi dari sebuah *dataset.*

Oleh karena itu, strategi umum yang diadopsi oleh banyak algoritma asosiasi adalah untuk menguraikan masalah yang terbagi kedalam dua sub-tugas utama [2] :

a. *Frequent Itemset Generation,* yang tujuannya untuk menemukan semua *itemset* yang memenuhi batas *minsup. Itemset* ini yang disebut sebagai *frequent itemset.*
b. *Rule Generation,* yang tujuannya untuk melakukan ekstraksi semua *rule* dengan nilai *confidence* yang tinggi dari *frequent itemset* yang ditemukan dari langkah sebelumnya. *Rule* ini disebut sebagai *rule* yang kuat. Persyaratan komputasi untuk *frequent itemset generation* biasanya lebih mahal dibandingkan *rule generation.*

2.2.3. *Market Basket Analysis*

Market Basket Analysis (MBA) sesuai namanya merupakan teknik data mining yang dapat diterapkan khususnya pada bidang seperti

pemasaran dan retail yang memiliki *dataset* keranjang belanja. Metode ini menguji pola pembelian pelanggan dengan mengidentifikasi hubungan asosiasi di antara berbagai item yang ditempatkan pelanggan di keranjang belanja mereka. Identifikasi asosiasi ini dapat membantu pihak retail memperluas strategi pemasaran dengan memperoleh wawasan tentang *item* yang sering dibeli bersama oleh pelanggan [14]. Hal ini sangat membantu untuk memeriksa perilaku pembelian konsumen dan membantu meningkatkan penjualan dengan fokus pada data transaksi penjualan.

Pada kasus retail, *Market Basket Analysis* melibatkan semua data retail pada transaksi pelanggan. Hasil ini akan membimbing mereka untuk merencanakan strategi pemasaran dan periklanan. Sebagai contoh, m*arket basket analysis* juga akan membantu manajer dalam mengusulkan pengaturan penempatan *item* di sebuah toko. Berdasarkan analisis ini, item yang secara teratur dibeli bersama bisa ditempatkan secara berdekatan dengan tujuan mempromosikan penjualan item secara bersama [3]. Jika konsumen yang membeli komputer juga cenderung membeli perangkat lunak anti-virus pada saat bersamaan, lalu penempatan *hardware* dekat dengan produk *software* akan membantu meningkatkan penjualan keduanya.

Bentuk masukan untuk *market basket analysis* adalah sebuah *dataset* transaksional atau keranjang belanja. Data ini memuat setiap *item* yang dibeli dalam satu kali transaksi. Atribut yang paling signifikan adalah identifikasi transaksi dan identifikasi *item* dengan mengesampingkan data kuantitas dan harga dari setiap *item.* Setiap transaksi menandakan satu perilaku jual-beli yang dapat dihubungkan ke identitas konsumen.

Tabel 2.2 Contoh Data Transaksi Keranjang Belanja

TID	Item
1	{Roti, Susu}
2	{Roti, Popok, Sirup, Telur}
3	{Susu, Popok, Sirup, Soda}

4	{Roti, Susu, Popok, Sirup}
5	{Roti, Susu, Popok, Soda}

Pada Tabel 2.2 merupakan contoh data dalam satu *dataset* transaksional yang besar. TID menunjukkan sebuah identitas unik atau sebuah nomor *invoice/*transaksi, dan pada kolom berikutnya adalah daftar *item* yang dibeli oleh pelanggan pada satu waktu transaksi. Dalam *market basket analysis* beberapa *rule* dapat diekstraksi dari *dataset* di atas, contohnya adalah {Roti} → {Susu} yang dapat dibaca setiap pelanggan yang membeli roti juga membeli susu. *Rule* ini menandakan hubungan yang kuat karena banyak pelanggan yang membeli kedua item ini secara bersamaan.

Tabel 2.3 Representasi Data Keranjang Belanja Dalam Biner

TID	Roti	Susu	Popok	Sirup	Telur	Soda
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Pada Tabel 2.3 menunjukkan data keranjang belanja juga dapat direpresentasikan di dalam bentuk biner dimana setiap baris menggambarkan transaksi dan setiap kolom menggambarkan item yang dibeli. Item dapat diperlakukan sebagai variabel biner yang bernilai 1 jika item ada dan bernilai 0 jika item tidak ada. Representasi ini merupakan pandangan yang paling sederhana tentang data keranjang belanja karena hanya memperhatikan aspek ada atau tidak adanya suatu *item* dan mengabaikan faktor lain seperti kuantitas dan harga item tersebut.

Perhitungan *market basket analysis* dimulai dari sebuah set item $I = \{i_1, i_2, ..., i_d\}$ di dalam data keranjang belanja dan $T = \{t_1, t_2, ..., t_N\}$ adalah set seluruh transaksi. Setiap transaksi t_i mengandung sub set item terpilih dari *I*. *Itemset* adalah kumpulan item yang terpilih. Jika sebuah *itemset* mengandung *k* item, maka dia disebut sebuah *k-itemset.* Sebagai contoh,

{Sirup, Popok, Susu} adalah sampel dari 3-*itemset. Null set* adalah *itemset* yang tidak memiliki item apapun.

Ada dua isu utama yang perlu diperhatikan saat menerapkan metode *market basket analysis* [14]. Pertama, menemukan pola dari kumpulan *dataset* transaksi yang besar secara komputasional bersifat mahal. Kedua, beberapa pola yang ditemukan berpotensi palsu karena bisa terjadi secara kebetulan dan tidak sengaja. Karena itu sebuah algoritma yang dipakai harus dapat mencegah kedua masalah ini.

2.2.4. Algoritma Apriori

Algoritma apriori diusulkan oleh Agrawal & Srikant pada tahun 1994 dan merupakan salah satu algoritma yang digunakan pada analisis asosiasi dan *market basket analysis.* Algoritma apriori melakukan banyak pencarian dalam *database* untuk menemukan *frequent itemset* dimana k-*itemset* digunakan untuk mengambil k+1-*itemset.* Pertama kali apriori melakukan pencarian 1-*itemset* yang digunakan untuk menemukan *itemset* dalam 2-*itemset* yang kemudian akan digunakan lagi untuk menemukan 3-*itemset* sampai pada batas maksimum jumlah k-*itemset* yang ditentukan [15].

Apriori yang mengambil semua transaksi di dalam basis data ke sebuah akun secara berurutan untuk mendefinisikan data transaksi. Data transaksi dapat direpresentasikan dengan aturan asosiasi (*association rule*), yang terdiri dari sisi kiri dan kanan (*Left => Right*). Sebagai contoh, sebuah *itemset* {A,B,C}dengan aturan {B,C} => {A} berarti “jika konsumen membeli *item* {B,C} maka dia kemungkinan akan membeli {A}”.

Untuk mengevaluasi aturan asosiasi yang dihasilkan, ada dua faktor yang dapat digunakan yaitu nilai *support* dan nilai *confidence.* Sebagai contoh {A,B} adalah sebuah *itemset* dan A=>B menjadi aturan asosiasinya. Nilai *support* adalah frekuensi relatif atau probabilitas P(A∩B), sementara nilai *confidence* adalah probabilitas kondisional dari B saat A terjadi,

P(B|A), yang sama dengan P(A∩B)/P(A). Aturan pada A=>B dengan nilai *support* 50% dan nilai *confidence* 85% berarti *item* A dan *item* B dibeli bersama dalam 50% transaksi dan 85% kasus yang membeli *item* B akan membeli *item* A juga P(A|B) [9].

Apriori mencari kombinasi item yang memenuhi syarat minimum nilai *support* dalam basis data. Nilai *support* didapatkan dari berapa jumlah *item* yang ditentukan oleh peneliti. Nilai *support* dari sebuah *item* diperoleh dengan Persamaan 1 :

$$Support\ (A) = P(A) = \frac{\textbf{Jumlah transaksi mengandung A}}{\textbf{Total Transaksi}} \quad \text{..... (1)}$$

Penentuan nilai *support* pada lebih dari satu *item* tetap menggunakan Persamaan 1 dengan modifikasi pada pembilangnya. Sebagai contoh nilai *support* dari dua *item* diperoleh dengan menggunakan Persamaan 2 :

$$Support\ (A,B) = P(A \cap B) = \frac{\textbf{Jumlah transaksi mengandung A dan B}}{\textbf{Total Transaksi}} \quad \text{..... (2)}$$

Frequent itemset menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (Φ). Jika nilai minimum (Φ) yang ditetapkan adalah 3, maka semua item dengan kemunculan lebih atau sama dengan 3 kali dapat disebut *frequent.* Himpunan dari *frequent* k-*itemset* dilambangkan dengan F_k.

Lift ratio merupakan salah satu faktor yang dapat dilihat untuk menentukan kuat tidaknya suatu aturan asosiasi. Sebuah *rule* dapat dikatakan sebagai *rule* yang kuat jika nilai *lift ratio* semakin mendekati nilai 1 . *Lift ratio* dihitung dengan berdasarkan nilai *confidence* dengan *expected confidence* [16]. *Confidence* dapat dihitung dengan Persamaan 3 :

$$Confidence(A,B) = P(B|A) = \frac{\textbf{Jumlah transaksi mengandung A dan B}}{\textbf{Jumlah transaksi mengandung A}} \quad \text{.....(3)}$$

Sementara nilai *expected confidence* dapat dihitung dengan Persamaan 4 :

$$Expected\ confidence = \frac{\textbf{Jumlah transaksi mengandung B}}{\textbf{Total Transaksi}} \quad \text{.....(4)}$$

Nilai *lift ratio* dapat dihitung dengan membagi nilai *confidence* dengan *expected confidence. Lift ratio* dapat dihitung menggunakan Persamaan 5.

$$Lift\ ratio = \frac{\boldsymbol{Confidence}}{\boldsymbol{Expected\ confidence}} \quad \text{.....(5)}$$

Nilai *lift ratio* yang lebih besar dari 0 dan mendekati nilai 1 menunjukkan bahwa suatu aturan asosiasi akan bersifat semakin kuat. Semakin tinggi *lift ratio*, maka semakin besar pula kekuatan asosiasinya.

Tingkat kekuatan sebuah algoritma dalam *association rule mining* dapat dihitung dan direpresentasikan dalam sebuah nilai akurasi. Tingkat akurasi dihitung dari hasil kali nilai *support* dan *confidence* semua *rule* yang dihasilkan dimana setiap *rule* yang terpilih harus memenuhi *lift ratio* yang telah ditentukan. Akurasi atau tingkat kekuatan apriori dapat dihitung menggunakan Persamaan 6 [9] :

$$\text{Nilai akurasi} = \frac{\sum_{i=1}^{n}(Si \; x \; Ci)}{n} \quad \text{.....(6)}$$

dimana :

n = jumlah aturan asosiasi

Si = nilai *support* untuk aturan asosiasi ke-*i*

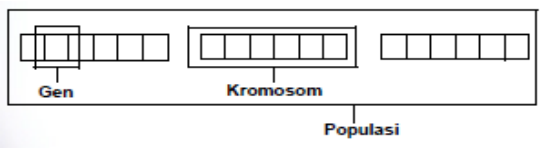
Ci = nilai *confidence* untuk aturan asosiasi ke-*i*

2.2.5. Algoritma Genetika

Algoritma genetika ditemukan oleh John Holland dan dikembangkan oleh David Goldberg. Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih alamiah. Salah satu aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi [8]. Algoritma genetika merupakan tipe algoritma optimasi yang berarti digunakan untuk menemukan solusi optimal pada masalah komputasional yang diberikan untuk memaksimalkan atau meminimalisasi fungsi tertentu.

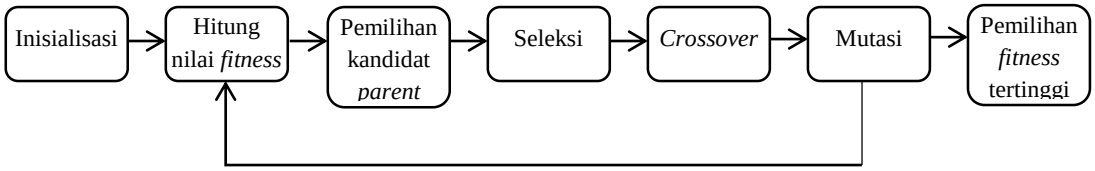
Satuan terkecil dalam algoritma genetika adalah sebuah gen. Gen menyatakan sebuah nilai dalam bentuk bilangan numerik, biner, simbol ataupun karakter yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Kromosom merupakan sebuah solusi yang

dibangkitkan dalam algoritma genetika, sedangkan kumpulan kromosom tersebut disebut sebagai populasi. Ilustrasi tentang ketiga elemen ini digambarkan pada Gambar 2.3



Gambar 2.3 Ilustrasi Gen, Kromosom dan Populasi

Algoritma genetika memiliki tujuh tahapan dimulai dari inisialisasi, kemudian proses utama dimulai penghitungan nilai *fitness* sampai pada proses seleksi [17]. Tahapan terakhir adalah seleksi semua kandidat berdasarkan nilai *fitness* tertinggi. Tahapan algoritma genetika ditunjukkan pada Gambar 2.4



Gambar 2.4 Tahapan Algoritma Genetika

Pada Gambar 2.4, tahap pertama adalah tahap inisialisasi adalah pembangkitan poulasi awal secara acak atau melalui prosedur tertentu. Pada tahap ini harus ditentukan berapa jumlah kromosom dalam satu populasi. Bentuk gen ditentukan dalam representasi biner, *real*, integer atau permutasi tergantung pada permasalahan yang ingin diselesaikan. Pada tahap inisialisasi juga ditentukan *treshold fitness*, jumlah iterasi maksimal, probabilitas *crossover*, dan probabilitas mutasi. Setelah semua ditentukan, kemudian populasi dibangkitkan secara acak.

Setelah populasi awal dibangkitkan, dilakukan penghitungan nilai *fitness* dengan fungsi *fitness* yang sesuai. Nilai *fitness* adalah fungsi yang menyatakan baik tidaknya suatu solusi dan dijadikan sebagai acuan dalam mencapai nilai optimal terhadap masalah yang diteliti [18]. Bentuk fungsi *fitness* dan fungsi objektif ditentukan berdasarkan masalah yang dihadapi. Nilai fungsi objektif dapat dihitung dengan menggunakan Persamaan 7 [19]:

$$f(x) = (a\ x\ supp(x)) + (b\ x\ conf(x)) \text{ ……..(7)}$$

dimana,

x = kromosom ke-x	*f*(x) = nilai fungsi objektif kromosom
supp(*x*) = nilai *support* kromosom	*a* = koefisien bobot *support* = 0,5
conf(*x*) = nilai *confidence* kromosom	*b* = koefisien bobot *confidence* = 0,5

Sedangkan nilai fungsi *fitness* untuk setiap kromosom (*association rule*) dapat dihitung dengan menggunakan Persamaan 8 [20] [21] :

$$fit(x) = \frac{1}{1+f(x)} \text{.....(8)}$$

Dengan keterangan,

fit(x) = nilai *fitness* untuk aturan ke-x

f(x) = fungsi objektif yang disesuaikan dengan permasalahan yang sedang dihadapi

Pada tahap pemilihan kandidat *parent* ditentukan jumlah pasangan *parent* yang diinginkan untuk dipilih kemudian dilakukan pemilihan dengan beberapa cara, antara ain adalah *random selection, tournament selection, rank selection* atau *fitness proportionate slection.*

Pada tahap seleksi dilakukan pemilihan calon induk untuk mendapatkan populasi selanjutnya. Seleksi dilakukan dengan menggunakan presentasi *fitness* setiap individu, dimana setiap individu mendapatkan luas bagian sesuai dengan presentase nilai *fitness*-nya. Proses seleksi menggunakan prinsip elitisme, dimana satu kromosom terbaik selalu disimpan selama proses sehingga akan selalu ada kromosom dengan nilai *fitness* tertinggi. Jika muncul kromosom baru dengan nilai yang lebih baik, kromosom yang lama akan ditukar dengan kromosom yang baru. Proses seleksi menggunakan beberapa parameter untuk pemilihan kromosom pada populasi selanjutnya, yaitu nilai probabilitas dan komulatif probabilitas. Probabilitas adalah suatu nilai yang digunakan untuk mengukur tingkat

terjadinya suatu kejadian yang acak. Untuk mencari probabilitas seluruh kromosom dapat menggunakan Persamaan 9 :

$$P(x) = \frac{fitness[x]}{total\ fitness} \quad \text{.....(9)}$$

dengan keterangan,

x = kromosom ke-x

fitness[x] = nilai *fitness* x

total fitness = jumlah total nilai *fitness* semua kromosom

Komulatif probabilitas adalah nilai komulatif dari total probabilitas dari nilai probabilitas yang pertama sampai index tertentu. Nilai komulatif probabilitas pada satu kromosom akan menunjukkan nilai total probabilitas sampai kepada nilai probabilitas pada kromosomnya. Untuk mencari komulatif probabilitas menggunakan Persamaan 10 :

$$C(x) = \sum_{k=1}^{x} P[k] \quad \text{.....(10)}$$

dengan keterangan,

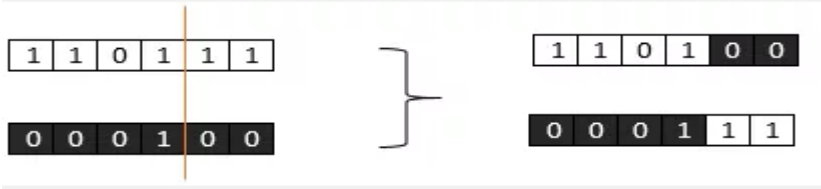
x = kromosom ke-x

C(*x*) = nilai komulatif x

P[*k*] = nilai probabilitas kromosom ke-k

Tahapan *crossover* merupakan tahap yang paling penting dalam algoritma genetika. Dalam tahap ini proses pembuatan anak atau kromosom baru dilakukan. *Crossover* melibatkan dua induk untuk menghasilkan keturunan yang baru. Proses ini dilakukan dengan pertukaran gen dua induk secara acak. Namun proses *crossover* tidak selalu dilakukan. Nilai acak dari kedua induk akan dibandingkan dengan probabilitas *crossover* nya. Jika nilainya lebih kecil atau sama dengan probabilitas maka akan dilakukan *crossover* dan sebaliknya. Ada beberapa jenis *crossover* yang dapat dipilih sesuai keperluan, yaitu :

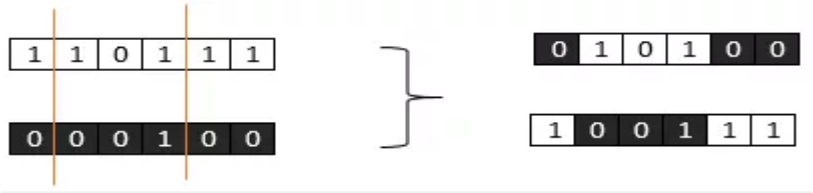
a. *One point crossover*



Gambar 2.5 *One Point Crossover*

Pada Gambar 2.5 *crossover* dilakukan dengan cara penukaran gen melalui satu titik *point* untuk menghasilkan kromosom baru.

b. *Multi point crossover*



Gambar 2.6 *Multi Point Crossover*

Pada Gambar 2.6 *crossover* dilakukan dengan cara penukaran gen melalui beberapa titik potong untuk membuat kromosom baru.

c. *Uniform crossover*

Gambar 2.7 *Uniform Crossover*

Pada Gambar 2.7 *crossover* dilakukan dengan cara penukaran gen melalui tiap indeks berdasarkan probabilitas. Sebagai contoh pada sebuah koin. Jika yang muncul kepala maka gen ditukar dan jika muncul ekor maka posisi gen tetap.

Pada tahap mutasi dilakukan penukaran nilai pada salah satu gen dengan nilai inversinya, misalnya gen yang bernilai 0 akan diubah menjadi 1. Setiap individu akan mengalami mutasi gen dengan probabilitas mutasi yang telah ditentukan. Proses mutasi ditunjukkan pada Gambar 2.8

Gambar 2.8 Proses Mutasi

Sama dengan tahap *crossover*, mutasi tidak selalu dilakukan. Jika nilai acak yang dimunculkan lebih kecil atau sama dengan probabilitas, maka mutasi akan dilakukan dan sebaliknya.

Tahap terakhir adalah tahap pemilihan final, yaitu ketika kondisi iterasi telah terpenuhi maka iterasi akan dihentikan. Nilai *treshold fitness* dan iterasi maksimal telah ditentukan pada tahap inisialisasi. Seluruh tahapan proses akan dihentikan jika salah satu kondisi ini telah terpenuhi. Seluruh kromosom akan diurutkan dan dilakukan pemilihan kromosom terbaik berdasarkan nilai *fitness* tertinggi.

2.2.6. TANAGRA

TANAGRA merupakan aplikasi *data mining* untuk tujuan akademis dan penelitian. TANAGRA memiliki beberapa metode *data mining* yaitu *exploratory data analysis, statisfical learning, machine learning* dan *database area.* Kelebihan dari aplikasi TANAGRA adalah kemudahan dalam menggunakannya, GUI yang mudah untuk dipelajari, memudahkan untuk analisa data asli maupun data sintetis, dan sifat *open source* sehingga peneliti yang menggunakan TANAGRA dapat mengakses *source code* dan menambahkan algoritmanya sendiri untuk memodifikasi fitur yang telah disediakan. Karena kelebihan ini, TANAGRA juga dapat dikategorikan sebagai alat untuk mempelajari teknik pemrograman khususnya dalam *data mining* [22].

2.2.7. Python

Python adalah bahasa interpreter yang berbasis objek dan bahasa pemrograman tingkat tinggi yang dapat digunakan untuk berbagai macam pengembangan perangkat lunak. Python menyediakan dukungan untuk integrasi dengan bahasa pemrograma lain [23]. Python dapat berjalan dibanyak platform/sistem operasi seperti Windows, Linux/Unix, Mac OS X, OS/2, Amiga, dan Palm Handhelds. Saat ini Python juga telah di porting kedalam mesin virtual Java dan .NET

Python didistribusikan dibawah lisensi OpenSource yang disetujui OSI (OpenSource Initiatives), sehingga Python bebas digunakan, gratis digunakan, bahkan untuk produk-produk komersil. Yayasan Perangkat Lunak Python – Python

Software Foundation (PSF) memegang dan melindungi hak atas kekayaan intelektual dibawah Python, tertuang dalam konferensi PyCon, serta mendanai proyek-proyek pada komunitas Python.