

## ANALISA POLA BELANJA MENGGUNAKAN ALGORITMA FP GROWTH, SELF ORGANIZING MAP (SOM) DAN K MEDOIDS

**Muhammad Imam Ghozali**

Fakultas Teknik, Program Studi Teknik Informatika  
Universitas Muria Kudus  
Email: imam.ghozali@umk.ac.id

**Ri'fan Zaenal Ehwan**

SMA N 1 Jakenan Pati  
Email : revan.ze@gmail.com

**Wibowo Harry Sugiharto**

Fakultas Teknik, Program Studi Teknik Informatika  
Universitas Muria Kudus  
Email: wibowo.harry@umk.ac.id

### ABSTRAK

Dalam sebuah bisnis, diperlukan upaya memaksimalkan keuntungan. Diantaranya dengan melakukan promosi. Ketepatan promosi dapat dipelajari dari database sebuah perusahaan ritel utamanya pola belanja pada produk yang biasa dibeli bersamaan. Informasi tentang pola belanja pelanggan yang tidak akurat menyebabkan kebijakan promosi tidak tepat dan efisien. Salah satu upaya lazim untuk memperoleh dan menggali pola belanja pelanggan adalah menggunakan data mining yang dikenal sebagai *Knowledge Discovery in Database* (KDD). Pendekatan yang biasa digunakan adalah asosiasi. Permasalahannya aturan asosiasi cenderung mengabaikan dataset yang besar. Untuk mengatasi hal tersebut dilakukan klasifikasi barang yang dibeli dan tidak dibeli bersamaan. Algoritma *Self Organizing Map* (SOM) dan *K-Medoids* cocok untuk diterapkan dalam mengcluster dataset besar. Penelitian ini menguji kevalidan dan kecepatan algoritma *Self Organizing Map* (SOM) dan *K-Medoids* jika dikombinasi dengan *Frequent Pattern-Growth* (FP-Growth).

**Kata kunci:** analisis keranjang pasar, rekomendasi produk, *self organizing map*, *k-medoids*, *fp-growth*.

### ABSTRACT

*In a business, it takes effort to maximize profits. Among them with promotions. The accuracy of the promotion can be learned from the database a major retail company spending patterns on products commonly purchased together. Information on customer shopping patterns inaccurate cause improper promotion policy and efisien. Salah a common effort to acquire and explore the shopping patterns of customers is using data mining known as Knowledge Discovery in Databases (KDD). The approach used is association. The problem tends to ignore the rules of association of large datasets. To overcome this problem do the classification of goods purchased and not purchased together. Algorithm Self Organizing Map (SOM) and K-Medoids mengcluster suitable to be applied in large datasets. This study tested the validity and speed algorithms Self Organizing Map (SOM) and K-Medoids when combined with Frequent Pattern-Growth (FP-Growth).*

**Keywords:** market basket analysis, recommendation product, promotion product, association rule mining, *k-medoids*, *fp-growth*.

### 1. PENDAHULUAN

Sebuah bisnis diperlukan upaya untuk memaksimalkan keuntungan. Keuntungan tersebut akan dapat diperoleh jika perusahaan dapat memanfaatkan potensi dan peluang usaha yang ada. Pelanggan/konsumen adalah sumber terbesar pendapatan sebuah usaha. Penjual dapat meningkatkan pendapatan dengan menggunakan strategi pemasaran yang tepat dan akurat. Salah satunya adalah menawarkan barang lain yang kemungkinan besar akan dibeli juga oleh pelanggan secara bersamaan dengan barang yang sudah direncanakan untuk dibeli sebelumnya. Permasalahannya adalah barang apakah yang hampir pasti dibeli oleh pelanggan dan dari manakah informasi barang rekomendasi tersebut kita peroleh? Database toko yang dimiliki toko tersebut dapat dianalisis sehingga memperoleh informasi mengenai produk yang

sering dibeli secara bersamaan. Metode yang lazim untuk memperoleh dan menggali pola pelanggan tersebut adalah dengan data mining yang biasa dikenal sebagai *Knowledge Discovery in Database* (KDD). Data mining dapat mengambil informasi yang penting dari database yang besar dan dibutuhkan manusia. Informasi atau pengetahuan tersebut menunjukkan konsep, alur, aturan maupun pola dan bentuk dari sekumpulan data yang berbeda-beda. Data mining dapat diterapkan dalam sebagian besar *field*. Misalnya untuk membangun trend, memprediksi masa depan, dan menganalisis faktor kunci yang dibutuhkan untuk sebuah permasalahan yang kompleks.

Banyak algoritma yang diusulkan untuk menemukan pola asosiasi dan algoritma pertama yang paling efisien untuk menemukan pola asosiasi adalah *apriori* [9][10]. Alasan utamanya karena *Frequent Pattern-Growth (FP-Growth)* merupakan algoritma yang dapat menghemat waktu dan media penyimpanan terlebih untuk database yang besar [12].

Algoritma Apriori adalah algoritma *association rule* konvensional, yang membutuhkan waktu yang sangat besar saat proses pemindaian database. Hal ini tentu saja berpengaruh pada efisiensi waktu dalam data mining. Algoritma yang diusulkan untuk mengatasi kelemahan tersebut adalah *Frequent Pattern-Growth (FP-Growth)* yang disebut yang tidak perlu untuk menghasilkan *generate candidate item sets* dan dianggap lebih efisien. *Frequent Pattern-Growth (FP-Growth)* dibangun dengan membaca kumpulan data satu transaksi pada waktu dan pemetaan setiap transaksi ke dalam bagian *Frequent Pattern-Tree (FP-Tree)* [2].

Permasalahan yang muncul selanjutnya dalam aturan asosiasi adalah masalah teknis yang berhubungan dengan teknik rekomendasi yang paling umum dimana aturan asosiasi cenderung mengabaikan *itemset* besar. Akibatnya rekomendasi *item* kurang tepat karena informasi tentang produk ritel tidak tersedia [8]. Untuk mengatasi masalah tersebut, atribut yang ada di *cluster* untuk membentuk kelompok atribut yang sama dan kemudian menentukan pola asosiasi pada masing-masing kelompok *Apriori*, sehingga dapat mempermudah proses mencari rekomendasi produk. Beberapa algoritma *clustering* diantara *K-Means*, *Fuzzy C-Means*, *Self Organizing Map (SOM)*, *K-Medoids* [9].

Dalam penelitian ini akan dikomparasikan kombinasi *Self Organizing Map (SOM)* dengan *Frequent Pattern-Growth (FP-Growth)*, *K-Medoids* dengan *Frequent Pattern-Growth (FP-Growth)* dan *Frequent Pattern-Growth (FP-Growth)* yang berdiri sendiri tanpa di *clustering* dengan algoritma apapun sebelumnya. Algoritma *Self Organizing Map (SOM)* dipilih karena *Self Organizing Map (SOM)* merupakan algoritma *clustering* model aglomerative dan partitif selain itu juga dikenal sebagai algoritma non liner, teratur, pemetaan dari input data dimensi tinggi ke dalam *array* dimensi rendah [4][5]. Algoritma *K-Medoids* dipilih sebagai perbandingan untuk mendampingi *Self Organizing Map (SOM)* dan *Frequent Pattern-Growth (FP-Growth)*. Pada dasarnya, algoritma *clustering* yang paling sering dipakai adalah *K-Means*. Namun, *K-Means* memiliki kekurangan yakni sensitif terhadap *outlier* dalam data dan hal ini bisa dijawab oleh *K-Medoids* yang tidak terpengaruh terhadap *outlier* data sehingga hasil *clustering* menjadi lebih akurat.

### 1.1 Analisis Pola Belanja Pasar

Membahas tentang pola belanja pelanggan tidak bisa dipisahkan dengan *Association Rule Mining* atau analisis keranjang pasar. *Association Rule Mining* merupakan salah satu metode data mining yang berfokus pada menemukan pola pembelian dengan mengekstraksi asosiasi atau kejadian dari data transaksional sebuah toko. Analisis keranjang pasar bermula dari transaksi-transaksi yang berisi satu atau lebih barang, dan beberapa informasi sementara dari transaksi tersebut. Untuk melakukan analisis keranjang pasar, berikut langkah-langkah:

- 1) Tentukan nilai *Minimum Support* yang diinginkan. *Minimum Support* merupakan ambang batas minimum jumlah *itemset* yang diperbolehkan, jika jumlah *itemnya* di bawah ambang batas maka *item* tersebut akan dieliminasi.
- 2) Menetapkan *frequent itemset* (kumpulan *item* yang muncul secara bersamaan), dengan cara mengambil *itemset* yang memiliki frekuensi *itemset* minimal sebesar *Minimum Support* sebelumnya.
- 3) Dari semua *frequent itemset*, hasilkan aturan asosiasi yang memenuhi nilai *Minimum Support*.

Analisis keranjang pasar didasarkan pada tiga matrik: *Support*, *Confidence* dan *Lift*. Ketiga matrik tersebut berasal dari catatan transaksi untuk bisnis [1].

#### *Support*

Matrik pertama ditetapkan untuk analisis keranjang pasar adalah *Support*, yang merupakan probabilitas dari asosiasi (probabilitas dari dua *item* yang dibeli bersama-sama). *Support* dihasilkan dari berapa kali jumlah *item A* dan *B* terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi tersebut. Untuk menghitung nilai *Support* digunakan persamaan (1) sebagai berikut:

$$\text{Support} = F(A \cap B) = \frac{\text{jumlah transaksi yang memuat A dan B}}{\text{total jumlah transaksi}} \quad (1)$$

#### Confidence

Confidence dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli. Untuk menghitung nilai Confidence digunakan persamaan (2) sebagai berikut:

$$\text{Confidence} = F(B/A) = \frac{\text{Support}(A \cap B)}{F(A)} \quad (2)$$

#### Lift

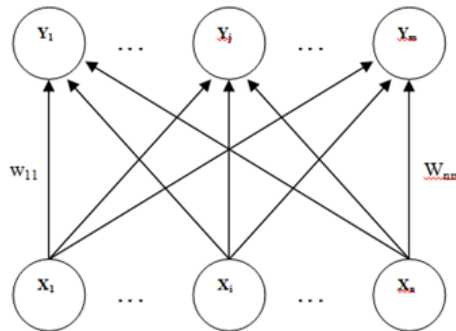
Lift Ratio mengukur seberapa penting rule yang telah terbentuk berdasarkan nilai support dan confidence. Lift Ratio merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B.

Untuk menghitung nilai Lift Ratio digunakan persamaan (3) sebagai berikut:

$$\text{Lift Ratio} = \frac{\text{Support}(A \cap B)}{\text{Support}(A) \times \text{Support}(B)} \quad (3)$$

### 1.2 Algoritma Self Organizing Map (SOM)

Self Organizing Map (SOM) merupakan salah satu metode dalam Jaringan Syaraf Tiruan (Neural Network) yang menggunakan pembelajaran tanpa pengarah (unsupervised learning). Metode pembelajaran tanpa pengarah tidak memerlukan target, selama proses pembelajaran unit – unit input yang hampir sama dikelompokkan dalam kelompok tertentu yang berada di unit output (cluster units). Arsitektur Self Organizing Map (SOM) terdiri dari 1 lapisan input dan 1 lapisan output. Setiap unit pada lapisan input dihubungkan dengan semua unit di lapisan output dengan suatu bobot keterhubungan yang ditunjukkan pada Gambar 1.



Gambar 1. Algoritma Self Organizing Map (SOM)

### 1.3 Algoritma Frequent Pattern-Growth (FP-Growth)

Algoritma yang biasa dipakai dalam mencari frequent item set antara lain algoritma Apriori dan algoritma Frequent Pattern-Growth (FP-Growth). Pada penelitian ini akan dibahas bagaimana pencarian frequent item set menggunakan algoritma Frequent Pattern-Growth (FP-Growth). Frequent Pattern Tree (FP-Tree) digunakan bersamaan dengan algoritma Frequent Pattern-Growth (FP-Growth) untuk menentukan frequent item set (data yang paling sering muncul) dari sebuah dataset. Algoritma Frequent Pattern-Growth (FP-Growth) adalah salah satu cara alternatif untuk menemukan himpunan data yang paling sering muncul tanpa menggunakan generasi kandidat [14]. Frequent Pattern-Growth (FP-Growth) membangun konstruksi data Frequent Pattern Tree (FP-Tree) yang sangat dikompresi, dan mengurangi data asli. Algoritma Frequent Pattern-Growth (FP-Growth) melakukan scan database yang sama sebanyak dua kali. Scanning database yang pertama, kita dapat memperoleh frequent 1-item-set, dan scanning database yang kedua, kita dapat memfilter database non-frequent item, selebihnya, Frequent Pattern Tree (FP-Tree) dihasilkan secara bersamaan. Akhirnya, dapat diperoleh aturan asosiasi dengan menggunakan Frequent Pattern Tree (FP-Tree) [13].

Kelemahan Apriori yang selalu melakukan scanning database secara berulang-ulang membuat Apriori ini kurang efektif. Berbeda dengan Frequent Pattern Tree (FP-Tree) yang digunakan bersamaan dengan algoritma Frequent Pattern-Growth (FP-Growth) yang hanya memerlukan dua kali scanning database untuk membuat frequent item set. Dengan menggunakan Frequent Pattern Tree (FP-Tree), algoritma Frequent Pattern-Growth (FP-Growth) dapat langsung mengekstrak frequent Item set dari

*Frequent Pattern Tree (FP-Tree)* yang telah terbentuk. *Frequent Pattern Tree (FP-Tree)* didefinisikan sebagai berikut [14] :

Sebuah *root* yang diberi label *null*, sekumpulan *sub-tree* yang beranggotakan *item-item* tertentu, dan sebuah tabel *frequent header*. Setiap simpul dalam *Frequent Pattern Tree (FP-Tree)* mengandung tiga *field*, yaitu:

- 1) *Item-name*: menginformasikan *item* yang dipresentasikan oleh simpul tersebut.
- 2) *Count*: mempresentasikan jumlah transaksi yang melewati simpul tersebut.
- 3) *Node-Link*: penghubung yang menghubungkan simpul-simpul dengan *item-name* yang sama, atau *null* jika kosong.

#### 1.4 Algoritma K-Medoids

Untuk melakukan *clustering* dengan metode partisi dapat menggunakan *K-Means* dan *K-Medoids*. *K-Means* merupakan suatu algoritma pengclusteran yang cukup sederhana yang mempartisi *dataset* kedalam beberapa *cluster* k. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan [14]. Kelemahan-kelemahan dari algoritma *K-Means* yaitu [20] :

- 1) Ketika jumlah data yang tidak begitu banyak, pengelompokan awal akan menentukan *cluster* secara signifikan.
- 2) Jumlah *cluster* K harus ditentukan terlebih dahulu.
- 3) *Cluster* yang asli tidak diketahui, dengan menggunakan data yang sama, jika dimasukkan dalam urutan yang berbeda dapat menghasilkan *cluster* yang berbeda jika jumlah data sedikit.
- 4) Kelemahan dari aritmatika mean tidak kuat untuk *outlier*, sangat jauh data dari *centroid* memungkinkan mempengaruhi *centroid* yang asli.

Algoritma *K-Medoids*, juga dikenal sebagai *partitioning around Medoids*, adalah varian dari metode *K-Means*. Hal ini didasarkan pada penggunaan *Medoids* bukan dari pengamatan *mean* yang dimiliki oleh setiap *cluster*, dengan tujuan mengurangi sensitivitas dari partisi yang dihasilkan sehubungan dengan nilai-nilai ekstrim yang ada dalam *dataset* [3]. Algoritma *K-Medoids* hadir untuk mengatasi kelemahan Algoritma *K-Means* yang sensitif terhadap *outlier* karena suatu objek dengan suatu nilai yang besar mungkin secara substansial menyimpang dari distribusi data [6].

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan jenis penelitian eksperimen, dengan melakukan ujicoba terhadap dataset yang telah dikumpulkan. Tahapan-tahapan penelitian ini adalah sebagai berikut:

- 1) Pengumpulan Data  
Pengumpulan data merupakan langkah awal pada suatu penelitian. Data yang digunakan pada penelitian ini adalah data transaksi penjualan Dr. Tariq Mahmood.
- 2) Pengolahan Data Awal  
Pengolahan awal (*Pre processing*) merupakan tahap untuk mempersiapkan data yang telah diperoleh dari tahap pengumpulan data, yang akan digunakan pada tahap selanjutnya. Pada tahap ini peneliti mengubah format data tanpa mengubah isi data agar mudah untuk diolah menggunakan aplikasi rapidminer.
- 3) Desain Eksperimen  
Tahapan ini akan membahas alurdesain eksperimen yang digunakan pada penelitian supaya prosesnya sistematis dan menghasilkan temuan yang valid.
- 4) Eksperimen dan Pengujian  
Tahapan ini akan mengeksekusi tahapan penelitian dengan teknik pengujian yang akan telah dipersiapkan sebelumnya.
- 5) Evaluasi Penelitian  
Tahapan ini akan membahas hasil evaluasi dari eksperimen yang telah digunakan.

Dataset ini merangkum 304 atribut dimana 303 diantaranya merupakan item barang yang dibeli. Salah satu datanya menunjukkan ID transaksi penjualan yang dilakukan dalam sebuah supermarket. Terdapat 1362 record transaksi dalam dataset ditunjukkan pada Gambar 2.

	A	G	H	I	J	K	L	M	N	O	P	Q
1	Basket ID	Sugar Cookies	Onions	Deli Ham	Dishwasher Detergent	Beets	40 Watt Lightbulb	Ice Cream	Cottage Cheese	Plain English Muffins	Strawberry Soda	Vanilla Ice Cream
2	C11867	false	false	false	false	false	false	false	false	false	false	false
3	C5096	false	false	false	false	false	false	false	false	false	false	false
4	C4295	false	false	false	false	false	false	false	false	false	false	false
5	C2837	false	false	false	false	false	false	false	false	false	false	false
6	C2693	false	false	false	false	false	false	false	false	false	false	false
7	C3497	false	false	false	false	false	false	false	false	false	false	false
8	C2696	false	false	false	false	false	false	false	false	false	false	false
9	C1895	false	false	false	false	false	false	false	false	false	false	false
10	C9524	false	false	false	false	false	false	false	false	false	false	false
11	C9527	false	false	false	false	false	false	false	false	false	false	false
12	C8726	false	false	false	false	false	false	false	false	false	false	false
13	C9383	false	false	false	false	false	false	false	false	false	false	false
14	C8582	true	true	false	false	false	false	true	false	false	false	false
15	C10133	false	true	false	false	false	false	false	false	false	false	false
16	C6983	false	false	false	false	false	false	false	false	false	false	false
17	C3641	false	false	false	false	false	false	false	false	false	false	false
18	C5246	false	false	false	false	false	false	false	false	false	false	false
19	C10136	false	false	false	false	false	false	false	false	false	false	false
20	C7787	false	false	false	false	false	false	false	false	false	false	false
21	C3644	false	false	false	false	false	false	false	false	false	false	false

**Gambar 2. Data Transaksi Penjualan**

Keterangan:

- 1) Basket ID seperti C11867, C5096, C4295, C2837, C2693, C2696, C1895, C9524 merupakan ID Transaksi penjualan barang.
- 2) Sugar Cookies, Onions, Deli Ham, Dishwasher Detergent, Beets, 40 Watt Lightbulb, Ice Cream, Cottage Cheese, Plain English Muffins, Strawberry Soda, Vanila Ice Cream dll merupakan barang-barang yang dijual di Supermarket.
- 3) False menandakan bahwa barang tersebut tidak dibeli oleh pembeli.
- 4) True menandakan bahwa barang tersebut dibeli oleh pembeli.

### 2.1 Pengolahan Data Awal

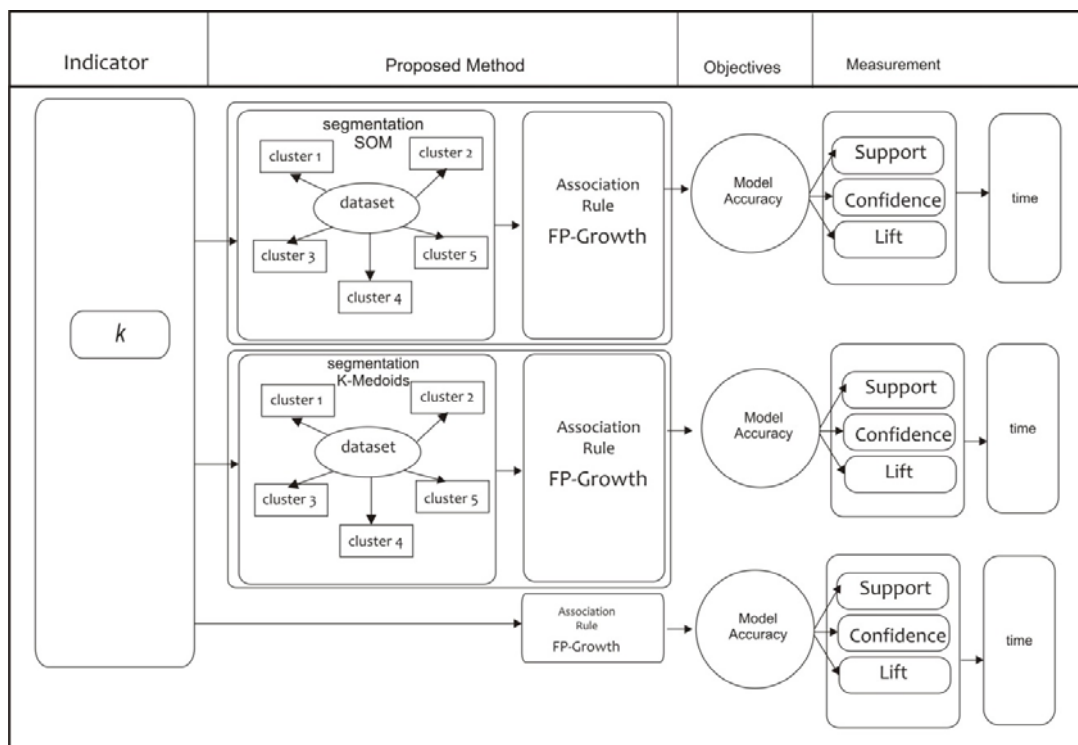
Data akan pada tahap pengolahan data awal diubah menjadi dataset yang sesuai dengan karakteristik algoritma yang akan dilakukan ujicoba. Diantaranya, menyesuaikan format *true* dan *false* dalam dataset menjadi data numerik 1 untuk *true* dan 0 untuk *false*, agar dapat digunakan dalam proses asosiasi. Tahapan selanjutnya melakukan *clustering* data tersebut tanpa menambah atribut apapun karena *Self Organizing Map* (SOM) akan meng*cluster* tersendiri. Sedangkan untuk *K-Medoids* ditentukan  $k = 5$

1	Basket ID	Hair Cond	Lemons	Standard	Frozen Cf	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets
2	C11867	0	0	0	0	0	0	0	0	0	0
3	C5096	0	0	0	0	0	0	0	0	0	0
4	C4295	0	0	0	0	0	0	0	0	0	0
5	C2837	0	0	0	0	0	0	0	0	0	0
6	C2693	0	0	0	0	0	0	0	0	0	0
7	C3497	0	0	0	0	0	0	0	0	0	0
8	C2696	0	0	0	0	0	0	0	0	0	0
9	C1895	0	0	0	0	1	0	0	0	0	0
10	C9524	0	0	0	0	0	0	0	0	0	0
11	C9527	0	0	0	0	0	0	0	0	0	0
12	C8726	0	0	0	0	0	0	0	0	0	0
13	C9383	0	0	0	0	0	0	0	0	0	0
14	C8582	0	0	0	0	0	1	1	0	0	0
15	C10133	0	0	0	0	1	0	1	0	0	0
16	C6983	0	0	0	0	0	0	0	0	0	0
17	C3641	1	0	0	0	0	0	0	0	0	0
18	C5246	0	0	0	0	0	0	0	0	0	0
19	C10136	0	0	0	0	0	0	0	0	0	0

Gambar 3. Dataset Setelah Diubah Menjadi Numerik

## 2.2 Desain Eksperimen

Eksperimen pada penelitian ini adalah menggunakan *Self Organizing Map* (SOM) dan *Frequent Pattern-Growth* (FP-Growth), kemudian dengan data yang sama menerapkan algoritma *K-Medoids* dan *Frequent Pattern-Growth* (FP-Growth) serta mengujinya juga dataset langsung diasosiasikan dengan *Frequent Pattern-Growth* (FP-Growth) tanpa diawali dengan *clustering* terlebih dahulu. Desain untuk eksperimen ini tunjukkan pada gambar 4.



Gambar 4. Desain Eksperimen Penggabungan Algoritma

Eksperimen ini diawali dengan menerapkan algoritma *Self Organizing Map* (SOM) untuk *clustering* dan algoritma *Frequent Pattern-Growth* (FP-Growth) untuk pendekatan asosiasi. Dalam Algoritma *Self Organizing Map* (SOM) dataset *dicluster* secara alamiah tanpa ditentukan dulu jumlahnya. Proses

*clustering* dilakukan supaya data yang diasosiasi menjadi lebih kecil sehingga pola yang diperoleh menjadi lebih akurat. Tiap *cluster* yang terbentuk kemudian diasosiasi dengan *Frequent Pattern-Growth (FP-Growth)* untuk membuat pola belanja pelanggan. Dalam proses asosiasi ini akan diukur tahapan, *confidence*, dan *lift ratio*. Setelah itu, dicatat waktu total yang dibutuhkan dari masing-masing tahapan. Eksperimen diulangi dengan mengganti algoritma *clustering Self Organizing Map (SOM)* dengan *K-Medoids* dan proses asosiasi tetap menggunakan *Self Organizing Map (SOM)*. Jumlah *cluster* yang dibuat yakni 5 *cluster*. Kemudian masing-masing *cluster* di asosiasi dengan algoritma *Self Organizing Map (SOM)* untuk diukur *support*, *confidence*, dan *lift ratio*. Kemudian waktu proses masing-masing tahapan dicatat. Akurasi dan total waktu proses *clustering* hingga asosiasi dari tahapan 1 yakni *Self Organizing Map (SOM) + Frequent Pattern-Growth (FP-Growth)* dibandingkan dengan tahapan 2 yakni *K-Medoids + Frequent Pattern-Growth (FP-Growth)* kemudian dibandingkan dengan akurasi dan waktu yang diperlukan jika menggunakan *Frequent Pattern-Growth (FP-Growth)* saja. Kemudian disimpulkan diantara ketiganya terkait akurasi dan efisiensi waktunya.

### 2.3 Evaluasi Penelitian

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Evaluasi rekomendasi produk untuk pelanggan ditunjukkan di gambar 5.

No.	Premises	Conclusion	Support	Confidence	Lift
1	Whole Corn	Hair Conditioner	0.970	0.976	1.006
2	Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
3	Whole Corn	Orange Flavored Fruit Bars, Hair Conditioner	0.970	0.976	1.006
4	Orange Flavored Fruit Bars	Whole Corn, Hair Conditioner	0.970	0.976	1.006
5	Whole Corn, Orange Flavored Fruit Bars	Hair Conditioner	0.970	0.976	1.006
6	Whole Corn	Orange Flavored Fruit Bars	0.994	1	1.006
7	Orange Flavored Fruit Bars	Whole Corn	0.994	1	1.006
8	Hair Conditioner	Whole Corn	0.970	1	1.006
9	Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
10	Hair Conditioner	Whole Corn, Orange Flavored Fruit Bars	0.970	1	1.006
11	Whole Corn, Hair Conditioner	Orange Flavored Fruit Bars	0.970	1	1.006
12	Orange Flavored Fruit Bars, Hair Conditioner	Whole Corn	0.970	1	1.006

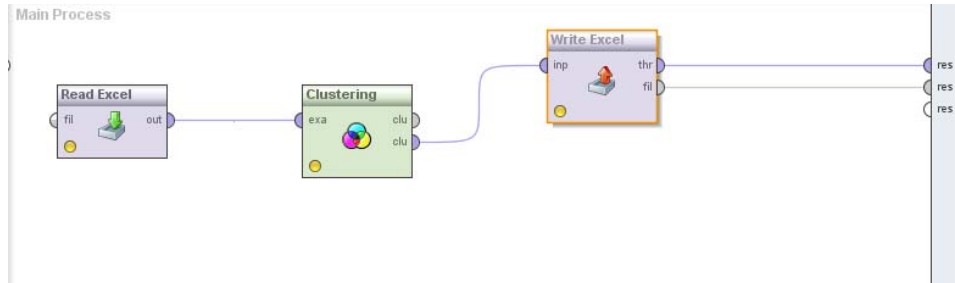
Gambar 5. Evaluasi Rekomendasi Produk Dengan *Support, Confidence, Lift*

Setelah ditemukan *Support, Confidence, Lift* kemudian dicatat pada masing-masing proses untuk diketahui seberapa efisien waktu yang diperlukan dari pengolahan dataset hingga menjadi sebuah pola belanja pelanggan. Kemudian akan disimpulkan tahapan mana yang memiliki tingkat kevalidan paling baik dan tahapan mana yang memiliki waktu proses paling efisien.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Clustering Data dengan K-Medoids

Dalam *k-medoids* kita tentukan menjadi 5 *cluster* (*k*). Penentuan *k = 5* ini dilakukan supaya lebih mudah didapatkan hasilnya karena keterbatasan kemampuan PC. Hasil dari proses *clustering* tersebut akan di ekspor dalam bentuk file terpisah untuk masing-masing hasil *cluster*. Proses *clustering* data ditunjukkan pada gambar 6, dan hasil *clustering* data ditunjukkan pada gambar 7.



Gambar 6. Proses Clustering Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R		
1	Basket ID	Cluster	ir	Conditio	Lemons	Standard	Frozen C	98pct. Fat	Sugar Coc	Onions	Deli Ham	Dishwash	Beets	40 Watt Lu	Ice Cream	Cottage C	Plain Engl	Strawben	Vanilla Ic	Pr
2	Basket ID	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
3	C5252	cluster_0	1	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0
4	C3659	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
5	C4478	cluster_0	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0
6	C887	cluster_0	0	0	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0
7	C7995	cluster_0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1
8	C4694	cluster_0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
9	C8987	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
10	C2177	cluster_0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0
11	C2543	cluster_0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0
12	C9233	cluster_0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
13	C7493	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	C7646	cluster_0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	C1784	cluster_0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0
16	C5303	cluster_0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0
17	C9623	cluster_0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	C959	cluster_0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0
19	C5501	cluster_0	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
20	C4706	cluster_0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
21	C12002	cluster_0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
22	C3770	cluster_0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0
23	C3788	cluster_0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
24	C6347	cluster_0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0

Gambar 7. Hasil Clustering Data

Setelah hasil *cluster* didapat, yakni *cluster 0*, *cluster 1*, *cluster 2*, *cluster 3*, *cluster 4* kemudian diurutkan/ sorting dan dipisahkan menjadi 5 file untuk kemudian di lakukan proses asosiasi dengan *Frequent Pattern-Growth (FP-Growth)*. Hasil dari *cluster* menggunakan *K-Medoids*:

*Cluster 0*: 32 items

*Cluster 1*: 2 items

*Cluster 2*: 2 items

*Cluster 3*: 2 items

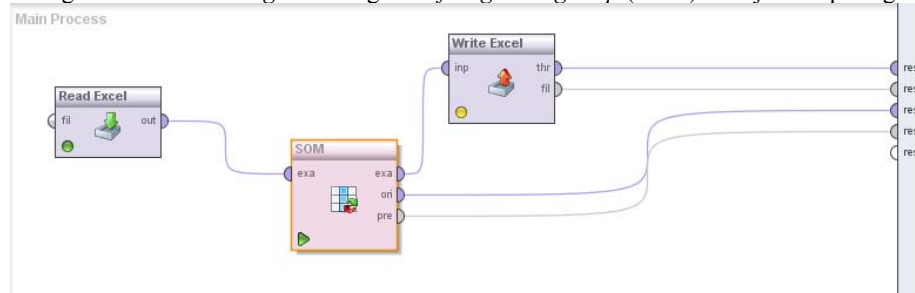
*Cluster 4*: 1323 items

Total number of items: 1361

Selama 28,20menit

### 3.2 Clustering Data dengan Self Organizing Map (SOM)

Rancangan Proses *clustering* data dengan *Self Organizing Map (SOM)* ditunjukkan pada gambar 8.



Gambar 8. Rancangan Cluster Data dengan Self Organizing Map (SOM)



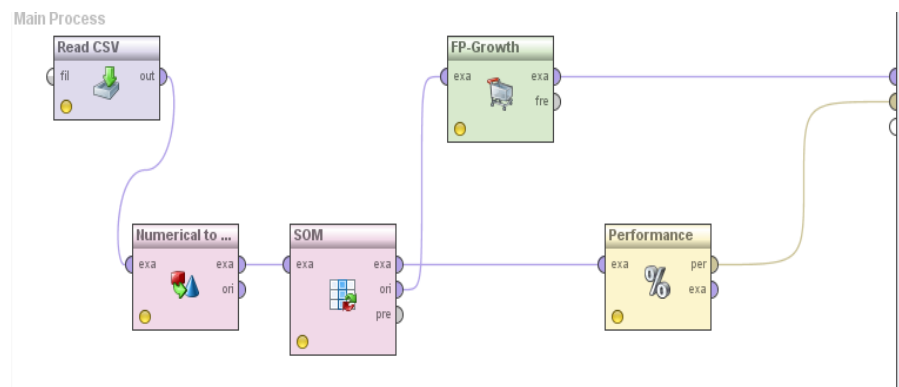
*Cluster* data dilakukan dengan *Self Organizing Map* (SOM) terbagi menjadi 2, dari rancangan *cluster* data tersebut didapatkan hasil yang ditunjukkan pada gambar 9.

Row No.	SOM_0	SOM_1
1	10	24
2	10	24
3	10	24
4	10	24
5	10	24
6	10	24
7	10	24
8	10	24
9	10	24
10	10	24
11	10	24
12	10	24
13	10	24
14	10	24
15	10	24

Gambar 9. Rancangan *Cluster* Data dengan *Self Organizing Map* (SOM)

### 3.3 Asosiasi *Cluster* Data Penjualan

Setiap *cluster* akan dilakukan asosiasi dengan *Frequent Pattern-Growth* (FP-Growth) untuk menghasilkan rekomendasi produk. Proses asosiasi *cluster* ditunjukkan pada gambar 10.



Gambar 10. Hasil Asosiasi *Cluster* Data Penjualan

Setelah proses *clustering* dilakukan, didapatkan 5 *cluster*. *Cluster* yang terbentuk dijadikan sebagai dataset baru yang lebih kecil daripada dataset sebelumnya. Kemudian setiap *cluster* akan dilakukan pendekatan asosiasi menggunakan algoritma *Frequent Pattern-Growth* (FP-Growth) untuk menganalisis keranjang pasar. Dengan dataset yang lebih kecil tujuan untuk meningkatkan akurasi dalam rekomendasi produk, sedangkan untuk perbandingan digunakan algoritma apriori.

## 4. KÉSIMPULAN

Hasil akurasi untuk menentukan rekomendasi produk kepada pelanggan yang didapat dari penelitian ini dapat diukur dengan menggunakan *Lift Ratio*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar produk A dibeli bersamaan dengan produk B. Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, produk A dan B benar-benar dibeli secara bersamaan [6]. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan *confidence*. Untuk eksperimen ditentukan nilai *Minimum Support* adalah 10% - 100% dan nilai *Minimum Confidence* 10% - 100%.

## DAFTAR PUSTAKA

- [1] B. Hoanca, "Using Market Basket Analysis to Estimate Potential Revenue Increases for a Small University Bookstore," *Information Systems*, vol. 1822, 2011, pp. 1-11.
- [2] C. Wang, W. Lee, and C. Pang, "Applying Fuzzy *FP-Growth* to Mine Fuzzy Association Rules," *Engineering and Technology*, 2010.
- [3] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*, Milan: WILEY, 2009.
- [4] Endy, Charles Lim, Kho I Eng, AntoSatriyoNugroho, 2010, *Implementation of intelligent searching Using self-organizing map for webmining used In document containing information in Relation to cyber terrorism, Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*.
- [5] Juha Vesanto and Esa Alhoniemi, Clustering of the Self-Organizing Map, *IEEE Transactions on Neural Networks*, 2000.
- [6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.
- [7] K. Christidis, D. Apostolou, and G. Mentzas, "Exploring Customer Preferences with Probabilistic Topics Models," *Work*, 2010, pp. 1-13.
- [8] L. Gordon, "Leading Practices in Market Basket Analysis," *Strategy*, vol. 94022, 2008.
- [9] M. Plasse, N. Niang, G. Saporta, A. Villeminot, and L. Leblond, "Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set," *Computational Statistics & Data Analysis*.
- [10] M. Ykhlef, "A Quantum Swarm Evolutionary Algorithm for mining association rules in large databases," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, 2011, pp. 1-6.
- [11] S. Guglani, G. Gupta, and M. Dhanda, "Way to Improve K-Means Algorithm by Using Various Attributes," (IJAEEST) *International Journal of Advanced Engineering Sciences and Technologies*, vol. 7, 2011, pp. 330 - 336.
- [12] Yongmei Liu and Yong Guan, "*FP-Growth* Algorithm For Application in Research of Market Basket Analysis," 2008, Capital Normal University, Beijing, China 2008.
- [13] Y. Liu and Y. Guan, "*FP-Growth* Algorithm for Application in Research of Market Basket Analysis," *Design*, 2008.
- [14] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. Mclachlan, A. Ng.