



## Analisis Pola Pembelian Pelanggan Menggunakan Algoritma Squeezer, Apriori dan FP-Growth Pada Toko Bangunan

Faris Syaifulloh<sup>1</sup>, Eva Yulia Puspaningrum<sup>2</sup>, M. Muharram Al Haromainy<sup>3</sup>  
<sup>1-3</sup> Universitas Pembangunan Negara Veteran Jawa Timur, Indonesia

Alamat: Jl. Rungkut Madya No.1, Gn. Anyar, Kec. Gn. Anyar, Surabaya, Jawa Timur 60294

Korespondensi penulis: [20081010078@student.upnjatim.ac.id](mailto:20081010078@student.upnjatim.ac.id)\*

**Abstract.** To compete with other stores, store owners need to design various strategies, one of which is understanding customer purchase patterns. This article examines the Squeezer algorithm and compares the performance of the Apriori and FP-Growth algorithms in forming customer purchase association patterns that can be used as a reference for store owners in planning sales strategies. The data mining process was carried out using Association Rules and Clustering methods. A total of 1256 sales transaction data samples were analyzed to understand the association patterns produced by each method. Based on the test results with a minimum support of 0.2 and a confidence of 0.6, the Apriori algorithm produced 194 association rules with a total rule strength of 1.16. Meanwhile, the FP-Growth algorithm produced 52 association rules with the same total rule strength of 1.16. The Clustering Method resulted in 7 clusters with a similarity value of 0.06322. After comparison, the FP-Growth algorithm proved to have better performance in generating association rules compared to the Apriori algorithm.

**Keywords:** Apriori, Association Rules, Clustering, FP-Growth, Squeezer.

**Abstrak.** Untuk dapat bersaing dengan Toko lain, pemilik toko perlu merancang berbagai strategi, salah satunya dengan mengetahui pola pembelian pelanggan. Artikel ini menguji algoritma Squeezer dan membandingkan kinerja algoritma Apriori dan FP-Growth dalam membentuk pola asosiasi pola pembelian pelanggan yang dapat dijadikan acuan bagi pemilik toko dalam merencanakan strategi penjualan. Proses data mining dilakukan menggunakan metode Aturan Asosiasi dan Clustering. Sebanyak 1256 sampel data transaksi penjualan dianalisis untuk memahami pola asosiasi yang dihasilkan oleh masing-masing metode. Berdasarkan hasil pengujian dengan minimum support sebesar 0,2 dan confidence sebesar 0,6, algoritma Apriori menghasilkan 194 aturan asosiasi dengan total kekuatan aturan sebesar 1,16. Sementara itu, algoritma FP-Growth menghasilkan 52 aturan asosiasi dengan total kekuatan aturan sebesar 1,16. Sedangkan pada Metode Clustering didapatkan 7 cluster dengan nilai similarity sebesar 0.06322. Setelah dilakukan Perbandingan, algoritma FP-Growth terbukti memiliki kinerja yang lebih baik dalam menghasilkan aturan asosiasi dibandingkan dengan algoritma Apriori.

**Kata kunci:** Apriori, Aturan Asosiasi, Clustering, FP-Growth, Squeezer.

### 1. LATAR BELAKANG

Perusahaan ritel termasuk toko bangunan, sering menghadapi tantangan dalam bersaing dan meningkatkan penjualan dan mempertahankan loyalitas pelanggan. Salah satu cara untuk mengatasi tantangan ini adalah dengan mencari dan memahami pola pembelian pelanggan (Aditiya *et al.*, 2020). Namun, analisis data transaksi penjualan yang besar dan kompleks memerlukan metode yang efisien dan akurat. Permasalahan penelitian ini adalah dengan bagaimana menemukan pola bagi pembelian pelanggan yang bermanfaat untuk strategi penjualan menggunakan algoritma data mining, yaitu Squeezer, Apriori, dan FP-Growth (Fikri Fajri, 2016).

Untuk mengatasi permasalahan tersebut, penelitian ini akan memanfaatkan data transaksi penjualan untuk menemukan pola pembelian pelanggan. Metode yang digunakan adalah Aturan Asosiasi dan Clustering. Algoritma yang akan diuji adalah Algoritma Squeezer, Apriori, dan FP-Growth. Sedangkan yang akan dibandingkan adalah Apriori dan FP-Growth (Firmansyah & Yulianto, 2021).

Penelitian ini bertujuan untuk Menganalisis dan membandingkan kinerja algoritma Squeezer, Apriori, dan FP-Growth dalam menemukan pola pembelian pelanggan di toko bangunan dan dapat mengidentifikasi aturan asosiasi yang dihasilkan oleh masing-masing algoritma serta memberikan rekomendasi strategi penjualan berdasarkan pola pembelian yang ditemukan (Ghozali *et al.*, 2017).

Dengan membandingkan algoritma Apriori dan FP-Growth serta menerapkan algoritma Squeezer, penelitian ini dapat bertujuan untuk memberikan materi pemahaman tentang pola pembelian pelanggan, sehingga pemilik toko bangunan dapat membuat keputusan yang lebih tepat dalam merencanakan strategi dalam pemasaran dan dalam manajemen stok barang. Berdasarkan penjelasan tersebut, penelitian yang dilakukan akan membandingkan hasil perhitungan algoritma FP-Growth dan Apriori serta mencari pola pembelian pelanggan menggunakan metode metode clustering dengan algoritma Squeezer dalam analisis pola perilaku pembelian pelanggan di toko bangunan (Gultom *et al.*, 2020).

## **2. KAJIAN TEORITIS**

Pengacuan pustaka merupakan elemen penting dalam penulisan ilmiah yang bertujuan untuk mengutip atau merujuk sumber informasi yang digunakan dalam karya tulis. Dengan melakukan pengacuan pustaka, penulis memberikan penghargaan kepada penulis asli atas ide atau data yang digunakan, serta menyediakan informasi yang memadai bagi pembaca untuk menemukan sumber aslinya. Pengacuan pustaka tidak hanya meningkatkan kredibilitas dan integritas karya tulis, tetapi juga memudahkan pembaca dalam melakukan verifikasi dan eksplorasi lebih lanjut mengenai topik yang dibahas.

### **2.1 Algoritma Apriori**

Algoritma Apriori melakukan eksplorasi terstruktur tanpa harus menjelajahi seluruh kandidat yang ada pada tahap pertama. Tahap kedua melibatkan ekstraksi aturan yang kuat. Frequent itemsets mengacu pada kombinasi item yang sering muncul bersama dalam data dan transaksi, seperti produk A dan produk B yang sering dibeli bersamaan dalam satu transaksi. Algoritma Apriori menggunakan pendekatan iteratif dengan pencarian tahap demi tahap, di mana k-itemsets digunakan untuk menemukan (k+1)-itemsets (Ikhwan *et al.*, 2015). Setelah

menemukan frequent itemsets, algoritma ini memanfaatkan pengetahuan dari frequent itemsets sebelumnya untuk menemukan informasi tambahan. Berikut adalah tahap utama dalam membentuk kandidat itemset dengan algoritma Apriori (Kamber & Pei, 2012).

1. Join (Penggabungan). Pada tahap ini, setiap item akan digabungkan dengan item lainnya sampai tidak ada lagi kombinasi yang dapat terbentuk.
  2. Pruning (Pemangkasan). Pada tahap ini, hasil kombinasi item yang terbentuk akan dipangkas berdasarkan nilai minimum support yang telah ditetapkan sebelumnya.
- Analisis asosiasi didefinisikan sebagai proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk support (dukungan minimum) dan confidence (kepercayaan minimum).

Adapun metodologi dasar analisis asosiasi terdiri dari dua tahap (Kamber & Pei, 2012):

## 2.2 Analisis pola frekuensi tinggi

Tahap ini bertujuan untuk menemukan kombinasi item yang memenuhi syarat minimum nilai support dalam basis data. Nilai support suatu item dihitung menggunakan rumus berikut.

$$\text{Support}(A) = \frac{\text{Jumlah Transaksi Mengandung } A}{\text{Total Transaksi}}$$

Selanjutnya untuk mencari nilai support dari gabungan 2 item dengan rumus berikut.

$$\text{Support}(A, B) = \frac{\text{Jumlah Transaksi Mengandung } A \text{ dan } B}{\text{Total Transaksi}}$$

## 2.3 Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, langkah berikutnya adalah mencari aturan asosiasi yang memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan asosiatif  $A \rightarrow B$ . Nilai confidence dari aturan  $A \rightarrow B$  dihitung menggunakan rumus berikut (Mantik *et al.*, 2022).

$$\text{Confidence } P(A/B) = \frac{\text{Jumlah Transaksi Mengandung } A \text{ dan } B}{\text{Transaksi Mengandung } A}$$

## 2.4 Algoritma FP-Growth

Algoritma FP-Growth adalah pengembangan dari algoritma Apriori yang bertujuan untuk menemukan frequent itemset (himpunan data yang sering muncul) dalam sebuah kumpulan data. FP-Growth dapat mengidentifikasi frequent itemset dengan akses yang lebih sedikit ke database asli, dan pendekatannya dianggap sebagai salah satu yang paling efisien dalam hal kinerja. Algoritma FP-Growth dirancang untuk mengatasi masalah yang muncul ketika jumlah kandidat itemset menjadi terlalu besar. FP-Growth menggunakan konsep

pembangunan pohon (tree) dalam pencarian frequent itemset, yang mengakibatkan kinerja algoritma ini lebih cepat dibandingkan dengan algoritma Apriori. Melalui pendekatan FP-Tree, algoritma FP-Growth mampu mengekstrak frequent itemset langsung dari struktur data FP-Tree. Proses pencarian itemset yang sering menggunakan algoritma FP-Growth dilakukan dengan cara membangun dan menggali struktur data pohon tersebut (Wang & Gao, 2021).

Penerapan Algoritma FP-Growth memiliki 3 tahap yaitu:

### **2.5 Pembangkitan Conditional Pattern Base**

Conditional Pattern Base adalah sub-database yang berisi prefix path (jalur awalan) dan suffix pattern (pola akhiran). Conditional Pattern Base dibangkitkan dari FP-Tree yang telah dibangun sebelumnya. Proses ini memungkinkan analisis yang lebih efisien dan terstruktur dalam menemukan pola-pola tersembunyi di dalam data transaksi.

### **2.6 Pembangkitan Conditional FP-Tree**

Pada tahap ini, jumlah support count dari setiap item dalam setiap conditional pattern base dihitung. Item-item yang memiliki jumlah support count yang lebih besar atau sama dengan minimum support count akan dibangkitkan menggunakan conditional FP-Tree.

### **2.7 Pencarian Frequent Pattern**

Pada tahap pencarian pola frekuensi, jika conditional FP-Tree berupa lintasan tunggal (single path), frequent pattern dapat ditemukan dengan mengombinasikan item-item dalam setiap conditional FP-Tree. Jika conditional FP-Tree bukan lintasan tunggal, maka proses pembangkitan dilakukan secara rekursif dengan menggunakan FP-Growth (Wang & Gao, 2021).

### **2.8 Algoritma Squeezer**

Algoritma Squeezer adalah salah satu algoritma yang digunakan dalam klusterisasi data. Squeezer memproses setiap pasangan data (tuple) dalam kumpulan data secara berurutan. Ketika membaca pasangan data pertama, algoritma akan membuat kelompok data (cluster) baru. Pasangan data berikutnya akan dimasukkan ke dalam kelompok data yang sudah ada atau ditolak oleh semua kelompok yang ada, sehingga membentuk kelompok baru berdasarkan fungsi kemiripan antara kelompok dengan pasangan data tersebut (Zengyou *et al.*, 2002).

Konsep algoritma Squeezer adalah mencari nilai kemiripan minimum antara suatu item dengan kluster yang ada. Jika nilai kemiripan minimum tersebut tidak terpenuhi, item tersebut akan dibuatkan kluster baru. Nilai kemiripan minimum menentukan seberapa mirip dua objek harus sebelum mereka dapat dianggap berada dalam kluster yang sama. Dengan menetapkan nilai kemiripan minimum yang ditentukan oleh pengguna, algoritma dapat mengontrol tingkat granularitas (kekasaran atau kehalusan) dari klusterisasi. Nilai Similarity Minimum yang

terlalu tinggi dapat menyebabkan kluster menjadi terlalu besar, sedangkan nilai yang terlalu rendah akan menghasilkan banyak kluster kecil yang mungkin terlalu spesifik. Rumus dari Algoritma Squeezer dapat dilihat seperti berikut (Zengyou *et al.*, 2002).

$$S(A, B) = X^{\alpha} Y^{\beta}$$

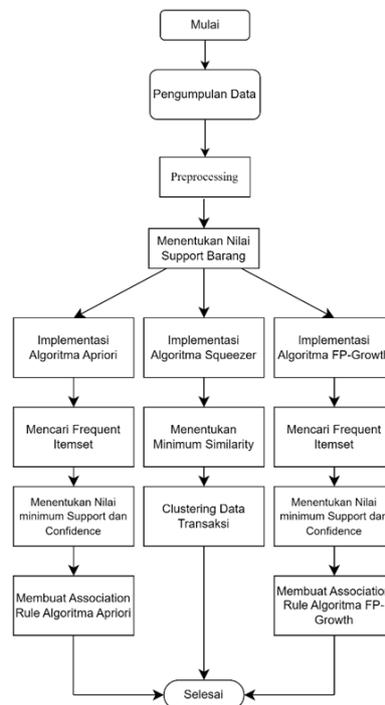
$$X = \frac{X(A, B)^2}{N(A) \times N(B)}$$

$$Y = \frac{N(A, B)}{N_{total}}$$

### 3. METODE PENELITIAN

Dalam Penelitian ini, terdapat beberapa tahapan yang dilakukan yaitu mengumpulkan data, preprocessing yang dilakukan seperti merubah data ke biner dan melakukan pengujian metriks terhadap nilai minimum support dan similarity dan melakukan pengujian terhadap algoritma Squeezer, Apriori dan FP-Growth. Berikut ini adalah tahapan dari penelitian.

#### 3.1 Pengumpulan Data



**Gambar 3.1 Alur Penelitian**

Data barang yang digunakan berasal dari dua jenis, yaitu data primer dan data sekunder. Data primer diperoleh langsung dari narasumber, berupa informasi transaksi di toko bangunan Sinar Pagi dalam bentuk file excel. Sementara itu, data sekunder merupakan data pendukung yang diperoleh dari sumber lain dan digunakan dalam penyusunan landasan teori untuk mendukung penelitian ini.

### 3.2 Preprocessing

Dilakukan untuk merubah dataset yang masih mentah menjadi dataset yang siap untuk diuji menggunakan algoritma

### 3.3 Merubah Format Dataset

Data yang semula mentah berupa struk nota penjualan berupa excel yang belum bisa dilakukan pengujian dengan algoritma secara langsung dan diperlukan preprocessing data seperti merubah format penulisan dataset menjadi CSV supaya sesuai format yang dibutuhkan oleh pengujian algoritma.

Data tersebut diubah menjadi file CSV dengan format "barang1;barang2", sesuai dengan struktur yang dapat diproses oleh algoritma Apriori, FP-Growth, dan Squeezer. Langkah ini memungkinkan persiapan data dalam format yang sesuai untuk penggunaan algoritma yang dibutuhkan.

**Tabel 3.1 Format Penulisan CSV**

1	"Kayu Meranti; Pipa PVC; Genteng"
2	"Pipa PVC; Kayu; Besi; Genteng"
3	"Asbes; Galvalum; Batu Bata"
4	"Cat Kayu; Waterproof; Triplek"
5	"Triplek; Galvalum; Waterproog"

Tabel 1. Di atas adalah contoh format penulisan data dalam bentuk CSV yang sebelumnya berasal dari file Excel. Data ini telah disesuaikan dan dipersiapkan untuk diuji menggunakan algoritma yang telah disebutkan sebelumnya. Format CSV ini memungkinkan data untuk diimpor dengan mudah ke dalam perangkat lunak analisis, sehingga memudahkan proses pengujian dan penerapan algoritma pada data yang telah disiapkan.

### 3.4 Membuat Struktur Data Biner

Data selanjutnya dikonversi menjadi data biner untuk dilakukan pengujian menggunakan library mlxtend untuk memproses data transaksi dan kemudian mengonversinya menjadi format DataFrame dengan menggunakan library pandas. Menggunakan Transaction Encoder digunakan untuk mengonversi data transaksi ke dalam format yang dapat diproses oleh algoritma.

```
from mlxtend.preprocessing import TransactionEncoder
te = TransactionEncoder()
te_data = te.fit(data).transform(data)
df =
pd.DataFrame(te_data, columns=te.columns_).astype(int)

df
```

**Gambar 3.2 Sintaks Proses Konversi Biner**

### 3.5 Metrik Pengujian

Metrik pengujian yang digunakan peneliti adalah nilai Support, Confidence, Lift dan Similarity Treshold.

Support digunakan untuk mengukur seberapa sering sebuah aturan muncul dalam dataset transaksi. Sebagai contoh, support dari aturan 'pasir  $\Rightarrow$  kerikil' adalah persentase transaksi yang mengandung kedua item tersebut dari total transaksi.

```
# Product Frequency / Total Sales
first = pd.DataFrame(df.sum() / df.shape[0], columns =
["Support"]).sort_values("Support", ascending = True)
first
```

**Gambar 3.3 Syntax Perhitungan Support**

Confidence mengukur seberapa sering aturan tersebut terbukti benar, yaitu seberapa sering item di sebelah kiri aturan muncul dalam transaksi yang mengandung item di sebelah kanan aturan. Sebagai contoh, confidence dari aturan 'pasir  $\Rightarrow$  kerikil' adalah persentase transaksi yang mengandung pasir juga mengandung kerikil dari total transaksi yang mengandung pasir.

Nilai confidence dihitung selama proses pembentukan aturan asosiasi menggunakan fungsi `Association_Rules` dari paket `mlxtend`. Setelah langkah-langkah selesai dan frequent itemsets ditemukan, aturan asosiasi dibentuk dan confidence dihitung untuk setiap aturan yang memenuhi ambang minimum yang ditetapkan, yaitu 40%.

```
# Association Rules & Info
df_ar = Association_Rules(freq_items, metric = "confidence",
min_threshold = 0.4)
df_ar
```

**Gambar 3.4. Syntax Perhitungan Confidence**

Lift mengukur seberapa besar ketergantungan antara dua item. Nilai lift lebih dari 1 menunjukkan adanya keterkaitan antara kedua item tersebut, sementara nilai kurang dari 1 menunjukkan bahwa item-item tersebut cenderung muncul secara independen.

Minimum similarity mengacu pada ambang batas minimum kesamaan atau kemiripan yang diperlukan untuk menyatukan dua data menjadi satu kluster. Perhitungan dari minimum similarity terjadi pada bagian di manafungsi `Squeezer` dipanggil untuk menghitung similarity antara item yang sedang dipertimbangkan (y) dengan setiap kluster yang sudah ada (clus).

#### 4. HASIL DAN PEMBAHASAN

Ketiga algoritma yang berbeda, yaitu Apriori, FP-Growth, dan Squeezer, diimplementasikan dalam penelitian ini. Algoritma Apriori menggunakan library Python 'mlxtend' yang menyediakan implementasi efisien. Penyesuaian parameter seperti nilai support minimum dan ukuran maksimum itemset dilakukan untuk mengatur kompleksitas dan relevansi pola yang dihasilkan.

```
# Apriori
freq_items = apriori(df, min_support = 0.2, use_colnames =
True, verbose = 1)
freq_items.sort_values("support", ascending = False)
```

**Gambar 4.5** Sintaks Perhitungan Algoritma Apriori

Selanjutnya, FP-Growth diimplementasikan menggunakan library yang sama dengan parameter yang serupa seperti Apriori, namun dengan pendekatan yang berbeda dalam pembentukan struktur data FP-Tree. Pendekatan ini bertujuan untuk mengurangi ketergantungan pada jumlah transaksi dalam proses analisis data.

Selanjutnya, algoritma Squeezer diterapkan menggunakan Library Python 'scikit-learn', yang menyediakan berbagai algoritma clustering termasuk k-means yang digunakan oleh Squeezer. Implementasi Squeezer melibatkan penyesuaian parameter seperti jumlah kluster yang diinginkan dan metode inisialisasi pusat kluster.

```
# Menerapkan algoritma FP-Growth untuk menemukan frequent
itemsets
frequent_itemsets_fp = fpgrowth(df_encoded, min_support=0.2,
use_colnames=True)
frequent_itemsets_fp
```

**Gambar 4.6** Sintaks Perhitungan Algoritma FP-Growth

```
def Squeezer(A, B, a = 1, b = 1):
    ke = secondIteration.index.get_loc((brg[B], brg[A]))
    na = sumbrg[A]
    nb = sumbrg[B]
    nab = secondIteration.freq[ke]
    x = pow(((nab*nab)/(na*nb)), a)
    y = pow((nab/n), b)
    hsl = x*y
    return hsl
```

**Gambar 4.7** Sintaks Perhitungan Algoritma Squeezer

Selain itu, nilai ambang kesamaan (similarity threshold) juga ditentukan untuk menentukan bagaimana dua data akan digabungkan menjadi satu kluster. Berikut ini adalah sintaks dari perhitungan squeezer.

#### 4.1 Hasil dari Algoritma Apriori

Dengan menggunakan minimum support sebesar 0,2% atau setara dengan minimal 20 kali pembelian dari total transaksi, peneliti berhasil mengidentifikasi 15 item yang sering muncul untuk satu itemset dan 97 item yang sering muncul untuk dua itemset. Tabel berikut menampilkan lima item dengan nilai support tertinggi untuk satu itemset.

Berdasarkan tabel diatas dapat disimpulkan bahwa lima item yang sering dibeli oleh pelanggan meliputi Waterproof, Asbes, Keramik, Cat kayu, dan Semen. Pada item pertama sekitar 47% dari total 1256 transaksi mengandung item Waterproof. Artinya, dari jumlah transaksi tersebut, sebanyak 594 transaksi termasuk dalam kategori pembelian Waterproof. Selanjutnya dilakukan proses pencarian minimal nilai support pada gabungan 2 itemset dan dihasilkan 97 item yang frequent.

**Tabel 4.2 Itemset dengan Support Tertinggi**

<i>No</i>	<i>Nama Item</i>	<i>Support</i>
1	Waterproof	0.47
2	Asbes	0.47
3	Keramik	0.47
4	Cat Kayu	0.45
5	Semen	0.44

Tabel diatas menunjukkan lima pasang item dengan nilai support tertinggi untuk dua itemset. Dalam pembelian dua item, terdapat lima pasang item yang sering dibeli oleh pelanggan, yaitu Waterproof dan Cat Kayu, Asbes dan Keramik, Waterproof dan Asbes, Keramik dan Genteng, serta Asbes dan Bata Ringan. Sebanyak 25% dari total transaksi mengandung pembelian antara Waterproof dan Cat Kayu, yang berarti dari 1256 transaksi, terdapat 314 transaksi yang mencakup pembelian kedua item tersebut.

**Tabel 4.3 Itemset dengan Support tertinggi**

<i>No</i>	<i>Nama item 1</i>	<i>Nama Item 2</i>	<i>Support</i>
1	Waterproof	Cat Kayu	0.25
2	Asbes	Keramik	0.24
3	Waterproof	Asbes	0.24
4	Keramik	Genteng	0.24
5	Asbes	Bata Ringan	0.24

Setelah dilakukan perhitungan dari support dan confidence, didapatkan 194 hasil Rule sebagai berikut.

**Tabel 4.4 Rule Pada Algoritma Apriori**

No	Nama Itemset	Support	Confidence	Lift
1	Cat Kayu => Waterproof	0.25	0.55	1.16
2	Waterproof => Cat kayu	0.24	0.52	1.16
3	Asbes => Keramik	0.24	0.52	1.11
4	Keramik => Asbes	0.24	0.52	1.11
5	Genteng => Keramik	0.24	0.53	1.14
6	Asbes => Waterproof	0.24	0.51	1.07
7	Keramik => Genteng	0.24	0.51	1.14
8	Waterproof => Asbes	0.24	0.51	1.07
9	Bata Ringan => Asbes	0.24	0.52	1.10
10	Asbes => Bata Ringan	0.24	0.50	1.10

Tabel diatas memuat 10 pola pembelian dengan nilai support tertinggi yang diidentifikasi menggunakan algoritma Apriori. Semua dari 10 aturan ini memiliki nilai lift lebih dari 1, menunjukkan bahwa terdapat keterkaitan yang signifikan antara item pertama dan item kedua dalam setiap aturan.

Pada Rule pertama menunjukkan pola pembelian pelanggan jika pelanggan membeli Cat Kayu, maka mereka juga cenderung membeli Waterproof. Ditemukan bahwa 25% dari keseluruhan transaksi melibatkan pembelian Cat Kayu dan Waterproof bersama-sama, serta bahwa 55% dari transaksi yang melibatkan pembelian Cat Kayu juga melibatkan pembelian Waterproof.

#### 4.2 Hasil Dari Algoritma FP-Growth

Algoritma kedua yang digunakan untuk menganalisis pola pembelian di toko bangunan Sinar Pagi adalah FP-Growth. Seperti halnya dengan algoritma Apriori, peneliti menetapkan minimum support sebesar 0,2% atau setara dengan minimal 20 kali pembelian dari total transaksi. Berdasarkan nilai support tersebut, ditemukan 15 item yang sering muncul dalam satu itemset dan 97 item yang sering muncul dalam dua itemset. Selanjutnya dilakukan proses pencarian minimal nilai support pada gabungan 2 itemset dan dihasilkan 97 item yang frequent.

**Tabel 4.5 Itemset dengan Support Tertinggi**

No	Nama item 1	Nama Item 2	Support
1	Asbes	Kerikil	0.23
2	Kerikil	Keramik	0.22
3	Pasir	Keramik	0.22
4	Waterproof	Cat Kayu	0.21
5	Waterproof	Kerikil	0.21

Tabel diatas menampilkan lima pasang item dengan nilai support tertinggi untuk dua itemset. Dalam pembelian dua item, terdapat lima pasang item yang sering dibeli oleh pelanggan, yaitu Kerikil dan Asbes, Kerikil dan Keramik, Kerikil dan Waterproof, Pasir dan Keramik, serta Waterproof dan Cat Kayu. Sebanyak 23% dari total transaksi mengandung pembelian antara Asbes dan Kerikil, artinya dalam 1256 transaksi, terdapat 288 transaksi yang mencakup pembelian kedua item tersebut, serta berlaku untuk kombinasi dua item berikutnya. Berdasarkan frequent itemsets yang telah didapatkan, terdapat 52 aturan asosiasi yang dihasilkan. Pada Tabel 4.9, disajikan 10 aturan dengan nilai support tertinggi.

**Tabel 6. Rule Pada Algoritma FP-Growth**

No	Nama Itemset	Support	Confidence	Lift
1	Waterproof => Bata Ringan	0.24	0.51	1.16
2	Bata Ringan => Waterproof	0.24	0.53	1.16
3	Galvalum=> Asbes	0.24	0.50	1.13
4	Asbes=> Galvalum	0.24	0.55	1.12
5	Genteng=> Keramik	0.23	0.50	1.14
6	Keramik => Genteng	0.23	0.52	1.12
7	Asbes=> Besi H Beam	0.23	0.51	1.15
8	Kerikil => Keramik	0.22	0.50	1.12
9	Pasir=> Keramik	0.22	0.50	1.11
10	Triplek => Pasir	0.21	0.50	1.14

Dari tabel diatas terlihat 10 pola pembelian dengan support tertinggi yang dihasilkan menggunakan algoritma Apriori. Dari 10 aturan yang ditemukan, semuanya memiliki nilai lift lebih besar dari 1, menunjukkan adanya hubungan yang signifikan antara item pertama dan item kedua dalam setiap aturan.

Pada aturan pertama, pola pembelian pelanggan yang ditemukan adalah JIKA pelanggan membeli Waterproof MAKA pelanggan cenderung juga membeli Bata Ringan. Ditemukan bahwa 24% dari total transaksi melibatkan pembelian bersama-sama antara Waterproof dan Bata Ringan, serta bahwa 55% dari transaksi yang melibatkan pembelian Waterproof juga melibatkan pembelian Bata Ringan.

#### 4.3 Hasil Dari Algoritma Squeezer

Sebelum melakukan clustering dengan algoritma Squeezer, dilakukan reduksi jumlah data dengan menggunakan minimum support sebesar 0,2. Setelah proses reduksi ini, jumlah item yang dikelompokkan berhasil dikurangi menjadi 97 item.

Minimum similarity yang ditetapkan oleh peneliti adalah 0,06. Berdasarkan nilai minimum similarity ini, terdapat 7 klaster yang terbentuk dari 97 item yang memenuhi syarat minimum similarity tersebut. Item-item yang tidak memenuhi kriteria minimum similarity di dalam satu klaster akan membentuk klaster baru.

**Tabel 4.2.** Hasil Cluster Algoritma Squeezer

Nama Barang	Cluster	Similarity
Asbes	Asbes	Baru
Besi H Beam	Asbes	0.06322
Cat Kayu	Asbes	0.06231
Galvalum	Asbes	0.06395
Keramik	Asbes	0.06576
Kerikil	Asbes	0.06111
Semen	Asbes	0.06012
Bata Ringan	Bata Ringan	Baru
Waterproof	Bata Ringan	0.06981
Genteng	Genteng	Baru
Kayu Meranti	Genteng	0.06017
Kabel	Kabel	Baru
Pasir	Pasir	Baru
Pipa PVC	Pipa PVC	Baru
Triplek	Triplek	Baru

Seperti yang terlihat dari Tabel diatas, dari 7 klaster yang terbentuk, hanya ada 3 klaster yang memiliki lebih dari satu anggota.

Dalam perhitungan dengan algoritma Squeezer, diterapkan minimum similarity sebesar 0,06. Setelah dilakukan perhitungan cluster oleh algoritma Squeezer, terlihat bahwa nama barang 'Asbes' belum memenuhi minimum similarity atau memiliki similarity di bawah 0,06. Hal ini menunjukkan bahwa 'Asbes' akan membentuk klaster baru.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah dilakukan analisis dan pembahasan pada Bab IV, berikut adalah kesimpulan yang diperoleh dari analisis perilaku pelanggan di toko bangunan Sinar Pagi.

1. Pada algoritma Apriori, ditemukan pola pembelian pelanggan bahwa jika pelanggan membeli Cat Kayu, maka mereka cenderung juga membeli Waterproof di toko bangunan.
2. Berdasarkan algoritma FP-Growth, ditemukan pola pembelian pelanggan bahwa jika pelanggan toko bangunan membeli Asbes, maka mereka juga cenderung membeli Galvalum.

3. Dengan menggunakan algoritma Squeezer, ditemukan pola pembelian bahwa item Waterproof dan item Bata Ringan sering dibeli secara bersama-sama oleh pelanggan toko bangunan.
4. Hasil perbandingan menunjukkan bahwa algoritma FP-Growth adalah yang terbaik

## 5.2 Saran

Berdasarkan analisis yang telah dilakukan, terdapat beberapa rekomendasi untuk mengembangkan toko bangunan:

1. Tempatkan item yang sering dibeli bersamaan oleh pelanggan di lokasi yang berdekatan untuk meningkatkan penjualan, serta berikan promo untuk barang-barang yang sering dibeli oleh pelanggan.
2. Untuk menemukan item yang sering dibeli bersama, metode Association Rule lebih disarankan dibandingkan metode clustering. Hal ini karena metode clustering membatasi setiap barang hanya dapat dimasukkan ke dalam satu cluster saja.

## 6. DAFTAR REFERENSI

- Aditya, R., Defit, S., & Nurcahyo, G. W. (2020). Prediksi tingkat ketersediaan stock sembako menggunakan algoritma FP-growth dalam meningkatkan penjualan. *Jurnal Informatika Ekonomi Bisnis*, 2, 67–73.
- Fikri Fajri, A. (2016). Implementasi algoritma Apriori dalam menentukan program studi yang diambil mahasiswa. *Jurnal Iptek Terapan*, 10(2), 81–85.
- Firmansyah, F., & Yulianto, A. (2021). Market basket analysis for books sales promotion using FP Growth algorithm, case study: Gramedia Matraman Jakarta. *Journal of Informatics and Telecommunication Engineering*, 4(2), 383-392.
- Ghozali, M. I., Ehwan, R. Z., & Sugiharto, W. H. (2017). Analisa pola belanja menggunakan algoritma FP Growth, Self Organizing Map (SOM) dan K Medoids. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 8(1), 317-326.
- Gultom, D. K., Arif, M., & Muhammad Fahmi. (2020). Determinasi kepuasan pelanggan terhadap loyalitas pelanggan melalui kepercayaan dedek. *MANEGGGIO: Jurnal Ilmiah Magister Manajemen*, 3(2), 273–282.
- Ikhwan, A., Nofriansyah, D., & Sriani. (2015). Penerapan data mining dengan algoritma FP-growth untuk mendukung strategi promosi pendidikan (Studi Kasus Kampus STMIK Triguna Dharma). *Jurnal Ilmiah SAINTIKOM*, 14(3), 211-226.
- Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.).
- Mantik, J., Nababan, A. A., Khairi, M., & Harahap, B. S. (2022). Implementation of K-Nearest Neighbors (KNN) algorithm in classification of data water quality. *Jurnal Mantik*, 6(1), 30–35.

- Rerung, R. R. (2018). Penerapan data mining dengan memanfaatkan metode association rule untuk promosi produk. *Jurnal Teknologi Rekayasa*, 3(1), 89.
- Wang, H., Bin, G., & Zheng, Y. (2021). Research on parallelization of Apriori algorithm in association rule mining. *Procedia Computer Science*, 183, 641-647.
- Zengyou, H., Xiaofei, X., & Shengchun, D. (2002). Squeezer: An efficient algorithm for clustering categorical data. *Structure*, 17(5).