



IMPLEMENTASI ALGORITMA *FP-GROWTH* UNTUK OPTIMALISASI STRATEGI PEMASARAN DI TOKO PAKAIAN: STUDI KASUS TOKO TREND

Batara Mahardika Aryoko¹, Saiful Nur Budiman², Sri Lestanti³

^{1,2,3} Teknik Informatika, Universitas Islam Balitar
 Blitar, Jawa Timur, Indonesia 66137

mahardsyech@gmail.com, sync.saifulnb@gmail.com, lestanti85@gmail.com

Abstract

TREND store is a retail outlet that sells various types of clothing and accessories. As market competition intensifies, the store needs to develop more efficient marketing strategies to remain competitive. One approach is to utilise sales data to analyse consumer purchasing patterns, given that such data has not been optimally used previously. This study aims to identify purchasing patterns and generate association rules as a basis for marketing strategies using the FP-Growth algorithm. The algorithm was chosen because it can identify frequent itemsets without candidate generation, making it more efficient than other methods in market basket analysis. The research data consist of 64 sales transactions from March 2025. In addition to pattern discovery, lift ratios were calculated to measure the strength of relationships between items. The results show that FP-Growth successfully identified significant purchasing patterns and generated relevant association rules. Several rules have lift ratios above 1, such as 1.2472 and 1.1463 for the combination K7, K1, C5, indicating positive relationships. These findings can be used to develop more data-driven and efficient marketing strategies, such as placing related items together to encourage impulsive purchases, supporting product recommendations, promoting planning, and informing other marketing decisions.

Keywords: Data Mining, FP-Growth, Lift Ratio, Market Basket Analysis, Marketing Strategy

Abstrak

Toko TREND adalah sebuah toko ritel yang menjual berbagai macam pakaian dan aksesoris. Dalam persaingan pasar yang semakin ketat, toko TREND perlu menyusun strategi pemasaran yang lebih efisien agar dapat bertahan. Salah satu upaya yang dapat dilakukan adalah memanfaatkan data penjualan untuk analisis pola pembelian konsumen, mengingat data tersebut sebelumnya belum digunakan secara optimal. Penelitian ini bertujuan untuk menemukan pola pembelian dan menghasilkan aturan asosiasi sebagai dasar strategi pemasaran menggunakan algoritma *FP-Growth*. Algoritma ini dipilih karena mampu menemukan *frequent itemset* tanpa proses pengembangan kandidat, sehingga lebih efisien dibanding metode lain dalam analisis keranjang pasar. Data penelitian terdiri dari 64 transaksi penjualan pada bulan Maret 2025. Selain proses pencarian pola, dilakukan juga perhitungan *lift ratio* untuk mengukur kekuatan hubungan antar item. Hasil penelitian menunjukkan bahwa *FP-Growth* berhasil menemukan pola pembelian yang signifikan dan menghasilkan aturan asosiasi yang relevan. Beberapa aturan memiliki nilai *lift ratio* di atas 1, seperti 1,2472 dan 1,1463 pada aturan dari kombinasi K7, K1, dan C5, yang berarti menunjukkan adanya hubungan positif. Temuan ini dapat dimanfaatkan untuk menyusun strategi pemasaran yang berbasis data dan tentunya lebih efisien. Misalnya item K7, K1, C5 diletakkan berdekatan agar pembeli melakukan pembelian impulsif, atau digunakan untuk rekomendasi barang, dan strategi lainnya.

Kata kunci: Analisis Keranjang Pasar, Data Mining, *FP-Growth*, Lift Ratio, Strategi Pemasaran

1. PENDAHULUAN

Perkembangan teknologi di era modern berkembang dengan sangat pesat, fenomena tersebut memiliki dampak yang sangat luas di berbagai sektor. Seperti di sektor bisnis dan industri, teknologi modern ini telah meningkatkan efisiensi operasional dan produktivitas dalam sektor tersebut. Misalnya, dalam hal pengambilan keputusan, teknologi dengan mudahnya mengambil keputusan yang paling

efisien dari data yang diperoleh untuk menemukan pola yang tidak terlihat.

Saat ini, dalam sektor bisnis ritel persaingan perdagangan saat ini terjadi dengan sangat ketat [1] dan sedang dihadapkan oleh tantangan-tantangan untuk tetap relevan dan kompetitif. Bisnis ritel kini perlu gerak cepat untuk beradaptasi di tengah tantangan seperti digitalisasi, perubahan pola belanja, hingga regulasi yang semakin

kompleks agar tetap kompetitif. Di sisi lain industri ritel dan pusat perbelanjaan terus berkembang seiring dengan perubahan tren, teknologi, dan perilaku konsumen [2]. *Managing director CRSC*, Yongki Surya Susilo mengatakan untuk menghadapi perubahan itu jawabannya terletak pada wawasan yang tepat, solusi yang inovatif, dan strategi pemasaran yang cerdas [2].

Dengan situasi tersebut, toko dengan skala besar atau kecil harus bisa untuk beradaptasi secepat mungkin terhadap perubahan-perubahan preferensi dari konsumen dan dinamika pasar. Ada begitu banyak cara untuk mengatasi masalah ini. Contohnya dengan mengoptimalkan strategi pemasaran toko yang didukung oleh analisis data yang akurat dan komprehensif.

Toko TREND adalah salah satu toko yang bergerak di bidang ritel pakaian atau *fashion*, toko TREND ini mempunyai beberapa cabang, salah satunya berada di Kecamatan Kesamben. Dengan banyaknya cabang yang dikelola di berbagai daerah, persaingan pasar yang sengit dan tidak menentu, dan juga dengan pengaruh teknologi yang sedang pesat sekarang ini, tentunya toko TREND perlu menerapkan teknologi yang terbaru untuk memudahkan pembuatan strategi pemasaran yang optimal seperti penataan barang agar konsumen melakukan pembelian yang impulsif, pemilihan barang untuk paket *bundling*, dan strategi pemasaran yang lainnya. Salah satu teknologi yang bisa digunakan dalam kasus ini adalah teknologi *data mining* untuk membuat strategi pemasaran yang lebih optimal.

Banyaknya persaingan dalam dunia bisnis khususnya dalam industri, menuntut para pengembang untuk menemukan suatu strategi yang dapat meningkatkan penjualan dan pemasaran produk yang dijual, salah satunya adalah dengan pemanfaatan data penjualan [3]. Tantangan banyak perusahaan adalah bagaimana mengolah koleksi data yang besar untuk mengekstrak informasi penting [4].

Data mining adalah proses menggunakan metode statistik, matematika, dan kecerdasan buatan, ilmu komputer untuk mengekstrak informasi berharga, hubungan, dan pola yang tidak langsung terlihat dari data dalam jumlah besar [5], [6].

Data dalam jumlah besar akan diproses menggunakan teknologi yang disebut *data mining*, yang membantu bisnis mengubah data yang masih mentah menjadi informasi yang berguna dan dapat digunakan untuk membuat pilihan bisnis yang penting. Deskripsi, klasifikasi, pengelompokan, asosiasi, pengurutan, peramalan, dan prediksi adalah tujuh peran utama *data mining*. *Data mining* mempunyai tujuan sebagai *Explanatory* yaitu untuk menjelaskan beberapa kondisi terkait dengan suatu penelitian, *Confirmatory* digunakan sebagai mempertegas hipotesis, dan *Exploratory* yang berguna dalam menganalisis data untuk hubungan baru yang tidak diharapkan [7].

Salah satu algoritma dalam *data mining* yang sering dipakai adapun seperti algoritma *Apriori* atau pun algoritma *FP-Growth* [8]. Algoritma *Apriori* dikembangkan menjadi algoritma *FP-Growth* [9]. Jadi bisa dibilang algoritma ini adalah generasi ke dua dari salah satu algoritma asosiasi, yaitu algoritma *Apriori*. Algoritma ini diusulkan oleh Han tahun 2000 [10], Han menyebutkan bahwa algoritma ini sangat efektif karena untuk menyelesaikan tugas penambangan hanya membutuhkan dua kali *scan* basis data [10]. Maka dari itu pada penelitian ini, algoritma *FP-Growth* akan digunakan untuk mencari pola yang dihasilkan dari data yang sudah diperoleh.

Kekuatan aturan asosiasi yang ditetapkan akan dinilai menggunakan pengujian *lift ratio*. *Lift ratio* menggambarkan tinggi atau tidaknya kekuatan aturan atas kejadian *antecedent* dan *consequence* acak. Nilai *lift ratio* memiliki rentang nilai 0 hingga tak terhingga. Nilai tersebut menyatakan keterkaitan suatu produk. Apabila hasil yang diperoleh dari perhitungan memiliki nilai kurang dari atau sama dengan 1 maka artinya kombinasi item yang telah terpilih tidak akan teridentifikasi adanya keterkaitan antara *antecedent* dan juga *consequent* [8].

Pengimplementasian algoritma *FP-Growth* pada penelitian ini akan menggunakan bahasa *python*. *Python* adalah salah satu bahasa pemrograman tujuan umum, artinya, bahasa ini bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja [11]. *Python* dapat digunakan untuk skrip sistem, pemrograman komputer mikro, pemecahan persamaan aritmetika, pengembangan web, dan pengembangan perangkat lunak atau pembangunan aplikasi [12].

Berdasarkan uraian tersebut, studi yang akan dilakukan bertujuan untuk mengimplementasikan sistem yang berbasis algoritma *FP-Growth* untuk mencari tahu pola dari pembelian pelanggan di toko pakaian TREND. Hasil analisis yang diperoleh akan dimanfaatkan untuk menyusun strategi yang lebih relevan dengan kebutuhan dan preferensi pelanggan, sehingga dapat meningkatkan adaptasi toko TREND untuk lebih bersaing.

2. METODE PENELITIAN

2.1 Tinjauan Pustaka

Penelitian terkait algoritma *FP-Growth* ini sudah banyak dilakukan, namun dengan tujuan yang berbeda-beda, namun masih terkait dalam urusan bisnis atau pemasaran. Seperti penelitian yang dilakukan oleh [13], yang menerapkan algoritma *FP-Growth* untuk menentukan pola pembelian produk. Penelitian ini bertujuan untuk menghasilkan analisis beberapa rekomendasi produk dengan kepercayaan tertinggi yang nantinya dapat digunakan untuk mengatur kategori produk pada suatu rak toko.

Selanjutnya penelitian yang dilakukan oleh [14], yaitu penelitian tentang penggunaan algoritma *FP-Growth* untuk menentukan penjualan item yang populer. Hasil yang

diperoleh dari perhitungan *FP-Growth* akan dimanfaatkan untuk menyarankan kepada pelanggan tentang item apa yang bagus untuk dibeli dengan item tertentu. Hal ini bisa dijadikan salah satu strategi pemasaran yang bisa meningkatkan pembelian yang impulsif dari pelanggan toko.

Selain itu, peneliti [15], juga meneliti algoritma *FP-Growth* untuk menentukan paket promosi pakaian di sebuah perusahaan, penelitian ini bertujuan untuk menemukan sebuah aturan yang bisa dipakai untuk mengembangkan strategi yang meningkatkan penjualan pakaian, hasilnya ditemukan lima pola asosiasi yang selanjutnya dibuatlah aturan yang bisa dipakai untuk merancang paket promosi pakaian.

Peneliti [1] dalam penelitiannya juga menggunakan algoritma *FP-Growth* untuk pengoptimalan strategi pemenuhan persediaan stok barang. Dan hasil yang didapat menunjukkan bahwa penerapan algoritma ini mampu mengungkap beberapa pola asosiasi dan menghasilkan beberapa *rule*, yang setiap *rule* tersebut memiliki nilai *confidence* yang memenuhi kriteria sehingga pola tersebut dapat di analisa untuk melakukan analisa manajemen persediaan barang.

2.2 Lokasi Penelitian

Penelitian ini dilaksanakan pada toko pakaian TREND di cabang Kesamben yang beralamat di Jalan raya Kesamben, Kecamatan Kesamben, Kabupaten Blitar, Jawa Timur 66191 pada bulan November 2024 sampai Juni 2025.

2.3 Teknik Pengumpulan Data

1) Sumber Data

Data penelitian ini diperoleh dari *database* penjualan toko TREND yang menyimpan riwayat transaksi pelanggan atau data per-nota. Data tersebut mencakup informasi item yang dibeli.

2) Jenis Data

Data yang digunakan termasuk data sekunder karena diperoleh dari sistem yang sudah ada yaitu *database* toko TREND, bukan dari hasil observasi atau kuesioner.

3) Metode Pengumpulan Data

Data diambil dengan mengeksport dari *database* toko ke dalam format yang dapat diolah seperti *CSV*, atau *Excel*. Proses ini dilakukan dengan bantuan pihak pengelola toko yang memberikan akses ke *database* penjualan.

4) Wawancara

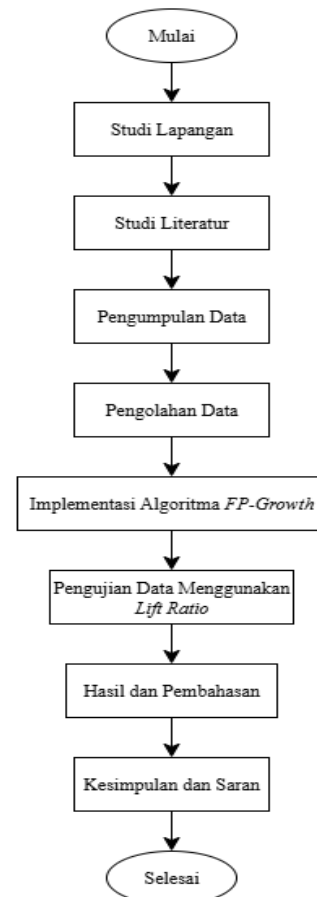
Wawancara juga dilakukan untuk menambah data yang diperlukan sehingga peneliti mempunyai wawasan yang lebih luas tentang tempat atau masalah yang diteliti pada penelitian ini.

5) Studi Pustaka

Studi pustaka juga dilakukan untuk belajar dan mencari informasi terkait penelitian yang sedang dilakukan agar menambah wawasan yang diperlukan.

2.4 Tahapan Penelitian

Tahapan penelitian terdiri dari enam langkah utama, yaitu studi lapangan, studi literatur, pengumpulan data, pengolahan data, implementasi algoritma, pengujian *lift ratio*, hasil dan pembahasan, dan yang terakhir adalah kesimpulan. Rangkaian tahapan ini digambarkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

1) Studi Lapangan

Untuk menggambarkan masalah, dan menentukan tindakan terbaik untuk menyelesaikan masalah yang ada, peneliti pertama-tama melakukan survei awal dengan tujuan memfasilitasi identifikasi masalah dengan memahami semua informasi yang termasuk dalam kasus industri ritel pakaian serta di toko pakaian TREND.

2) Studi Literatur

Peneliti memeriksa literatur yang relevan pada tahap kedua dari proses penelitian untuk mendapatkan pengetahuan dan wawasan yang akan membantu dalam mengidentifikasi

teori, pendekatan, atau strategi terbaik untuk menyelesaikan masalah yang muncul.

3) Pengumpulan Data

Pada tahap ini, peneliti melakukan pengumpulan data dengan pendekatan wawancara pada penanggung jawab toko pakaian TREND. Selain itu, peneliti juga mengambil data penjualan pada toko TREND dengan izin dan bantuan pengelola toko.

4) Pengolahan Data

Selanjutnya, peneliti melakukan pengolahan data dengan pembersihan dan menyiapkan data untuk dipakai lebih lanjut, pengolahan data ini termasuk menangani data yang hilang, mengubah format data jika diperlukan, dan menghapus data yang tidak diperlukan.

5) Implementasi Algoritma *FP-Growth*

Pada tahap ini, peneliti akan membahas bagaimana pengimplementasian algoritma *FP-Growth* dalam mencari pola item terbanyak dari *dataset* yang sudah disiapkan pada tahap sebelumnya. Pada tahap ini juga peneliti mengimplementasikan algoritma *FP-Growth* ke dalam aplikasi sederhana dengan memanfaatkan bahasa *python*.

6) Pengujian Data Menggunakan *Lift Ratio*

Setelah pengimplementasian algoritma *FP-Growth*, peneliti akan menguji data yang berbentuk pola item dari hasil perhitungan algoritma *FP-Growth* yang sudah dilakukan dengan menggunakan pengujian *lift ratio*.

7) Hasil dan Pembahasan

Pada tahap ini, peneliti akan menjelaskan tentang perhitungan algoritma *FP-Growth* yang sudah dilakukan dan aplikasi yang sudah dibuat agar para pembaca mudah memahami isi dari penelitian ini.

8) Kesimpulan dan Saran

Tahap akhir, peneliti menuliskan kesimpulan dan saran dari hasil yang diperoleh pada penelitian yang dilakukan dengan rinci agar pembaca dapat mengerti dan memahami tujuan dari penelitian.

3. HASIL DAN PEMBAHASAN

3.1 Data Transaksi

Pada penelitian ini, peneliti menggunakan data transaksi toko TREND pada bulan Maret 2025, terdapat kurang lebih 16.000 data transaksi pada bulan tersebut. Setelah data diperoleh, data akan diproses lebih lanjut agar bisa digunakan. Setelah diproses, terdapat 8.978 data yang memenuhi syarat untuk perhitungan. Namun untuk lebih memudahkan peneliti dan pembaca dalam memahami algoritma *FP-Growth*, dari ribuan data yang sudah didapatkan peneliti akan mengambil beberapa data seperti pada Tabel 1 yang nantinya akan dijadikan sampel untuk perhitungan pada penelitian ini.

Tabel 1. Sampel Data Transaksi

<i>TID</i>	Daftar Barang
1	C6, K7, S2, P2, S3, C5, T5, B6, K1, T4, S5
2	T6, S2, S4, K8, K1, J3, S3, T3, B6, G1, P5
3	B3, C6, H1, K1, R2, K3, K7, D4, B6, H2, G1
4	B4, S2, K2, K1, B3, C3, D4, K7, B6, T3, S4, G1, C4
5	K7, M1, C5, B6, C4, D4, B2, G1, T6, C2, S7, M2, P1
6	K7, B2, G1, M2, C5, S2, T4, S8, P3, J2
7	D4, M5, B6, CZ2, B3, S2, B1, T3, M2, P1, C9
8	K7, K1, C2, S6, M3, P5, C8, M2, J3, S7, P1
9	M2, G2, M1, C6, S5, D4, K1, C5, K7, S3
10	K1, C1, K7, R2, C6, C5, G1, M2, J3, P1
11	K1, C5, K7, K4, S2, S7, D3, M2, S6, A1, J3, D2
12	S2, M4, D4, K3, T6, K1, C6, B4, CZ2, R2, C5
13	K7, K3, C5, C4, K1, G1, C3, S2, J3, P1
14	B1, C2, C6, B6, G1, B3, C4, S5, K1, S2, A1, P5, K3
15	B2, K3, K7, S2, C6, C5, T5, S4, K5, M1
16	J2, C2, C9, C6, C7, C5, H2, K3, B3, D4
17	D4, M3, C6, K7, K3, B2, K1, C5, S2, G1, R2, C3, K6
18	M2, B3, K1, C5, B4, J4, D4, K3, K7, G1, B2
19	T3, S2, KZ1, T6, T5, D1, K7, B4, B3, C6, C2
20	J3, P5, P1, K1, S7, C5, S2, S4, P4, A1, D2
21	S7, K6, K7, C6, T1, S2, G1, C6, CZ2, R3
22	K5, C5, B2, B4, K7, C6, C5, P1, P5, S7
23	K1, C3, C2, T6, S2, B2, C6, K7, SZ1, K3, T3, S9, C5
24	M2, C2, B6, K1, D1, J1, R1, S2, C6, C4
25	C3, B4, K1, K7, B1, S2, C5, C6, K3, M1
26	P1, S7, M2, K1, C2, J2, K3, S2, R1, H1
27	K1, K3, C5, R1, S2, K7, C6, SZ1, T5, C2
28	S2, A1, P2, K1, K7, K3, C5, D4, C3, C1, R2, J3
29	D2, P1, S7, B6, K1, K7, B3, C7, D1, C4, R2
30	S2, K7, G1, D1, K4, K1, C5, C3, B2, B3, D4, S7, T5
31	C5, B2, K1, K3, B3, D4, D1, C6, C3, T5, S2, J2
32	S2, T3, K7, B4, C6, C5, C3, H2, K1, B2, K3
33	T6, B4, H2, C2, B6, G1, C3, K7, P1, M2
34	A1, C5, B2, C7, S7, H1, B3, T3, S5, K3, H2, R2, K1, D4, C6, M2
35	S4, P2, C6, B3, B6, K3, C5, B2, K7, M1, S5
36	C6, D4, G1, D1, B2, H2, K1, B4, P1, D2, D3, M2, S3

<i>TID</i>	Daftar Barang
37	S2, J3, S7, D2, M2, S1, H2, B4, K7, G1, R1
38	D1, C6, B4, C2, T2, K1, K8, M2, S6, S7
39	K7, H2, K1, S2, C4, K3, G1, R1, R2, C3
40	K7, SZ1, C5, B3, S5, K1, R2, C3, B6, D1
41	T3, K1, K4, C7, C3, C6, K3, K7, P5, M2, J3, S7
42	G1, C2, B3, D4, B2, K7, B8, P1, S7, K9
43	M4, C2, C3, C5, D4, G1, SZ1, P5, S7, K5
44	H2, S7, D2, K4, C3, R2, K7, P6, B3, G1
45	K5, P5, M2, S7, B6, K1, C2, B4, G1, C7, K7, SZ1, C5, C3, J1, M1, S2
46	M1, C7, K7, C3, SZ1, B3, K3, C6, K5, B2
47	K4, B4, K3, K7, C2, C5, B6, G1, K1, D1
48	C5, K1, R1, T3, H2, G1, K7, R2, B2, D4
49	C4, C5, C6, K7, B7, H2, P5, M2, C9, S7, K4, S2, S3
50	H2, C3, K7, C6, R2, D4, P5, S2, P2, S3
51	T3, H2, S2, B4, K3, C3, K1, C5, SZ1, H1
52	T6, H2, C6, K7, D4, S5, K2, C2, T5, C4, B3, G1, B5
53	G1, C5, K3, K7, B2, P6, D4, B3, S5, T3, S4, M2
54	K7, D4, G1, O1, T6, C6, K1, SZ1, C3, B3
55	K7, K2, T3, D4, S2, J1, S7, M2, P1, D2
56	T6, K7, S2, T3, S4, B2, C5, K1, G1, SZ1, C6, CZ1, B8
57	K1, C6, D1, B6, M4, C5, S3, K1, H1, C2, C3, K3
58	B6, B3, C6, R2, K7, S5, B2, G1, C2, C3, S2, P2
59	C7, C6, K4, K7, C5, T3, J1, K1, R2, P1, M2, M1
60	D4, B3, H1, K9, G1, C3, H2, R2, K4, B2
61	P1, G1, K1, R2, D4, C5, H2, B6, C2, B2
62	C2, P2, C5, K1, K3, M2, L1, K4, B6, C6
63	S4, M2, C2, C5, C4, B4, D4, G1, D1, B2
64	S2, K7, S7, C5, C4, T6, B2, C6, D4, K1

3.2 Menentukan *Support* dan *Confidence*

Setelah *dataset* sudah siap, langkah pertama perhitungan algoritma *FP-Growth* adalah menentukan *minimum support*. *Minimum support* digunakan sebagai ambang batas seberapa suatu item tersebut muncul agar bisa dianggap penting, sedangkan item yang tidak memenuhi akan dieliminasi. Pada penelitian ini peneliti akan menggunakan *minimum support* 15. Selain itu, *minimum confidence* juga ditentukan untuk menguji keterkaitan antar produk yang sudah ditemukan, dan pada perhitungan ini peneliti akan menggunakan *minimum confidence* 70% (0,7).

Terdapat beberapa cara untuk menentukan *minimum support* dan *minimum confidence*. Salah satunya adalah merujuk literatur dan studi sebelumnya, di mana nilai *minimum support* umumnya ditentukan antara 1% - 20% dan *confidence* di atas 50% dianggap cukup baik, namun penentuan nilai tersebut juga perlu disesuaikan berdasarkan banyaknya data dan tujuan analisis. Cara lainnya adalah melalui eksperimen, yaitu dengan mencoba beberapa nilai *minimum support*, dan nantinya akan dipilih yang paling sesuai. Nilai yang terlalu kecil biasanya akan menghasilkan terlalu banyak pola, sedangkan nilai *minimum support* yang terlalu besar dapat menghilangkan pola yang potensial. Dalam konteks *dataset* yang berjumlah 64 transaksi, *minimum support* dipilih agar pola yang terbentuk tetap relevan dan tidak terlalu sedikit.

3.3 Menentukan *Frequent Itemset*

Setelah ditentukannya *minimum support*, langkah selanjutnya adalah membentuk *itemset* dari *dataset* transaksi. Proses ini dimulai dengan membentuk 1-itemset, yaitu menghitung frekuensi kemunculan setiap item pada seluruh transaksi. Setiap item kemudian dihitung jumlah transaksinya (*support count*), lalu dibandingkan dengan nilai *minimum support* yang telah ditentukan sebelumnya, item yang memenuhi *minimum support* akan dipertahankan sedangkan item yang tidak memenuhi nilai tersebut akan dieliminasi. Tahap ini penting untuk memastikan bahwa hanya item yang benar-benar signifikan yang digunakan dalam proses pembangkitan pola selanjutnya.

Sebagai contoh, item seperti K7, K1, C5, dan C6 memiliki jumlah kemunculan di atas ambang *minimum support* 15, sehingga ketiganya masuk ke dalam *frequent 1-itemset*. Sebaliknya item lain yang muncul kurang dari 15 kali tidak disertakan karena tidak dianggap memberikan kontribusi yang berarti. *Frequent 1-itemset* inilah yang kemudian digunakan untuk menyusun transaksi terurut (*ordered transaction*) yaitu setiap transaksi disusun ulang berdasarkan urutan item dengan *support* terbesar. Transaksi terurut ini menjadi dasar dalam proses pembentukan *FP-Tree*.

Tahap penentuan *itemset* ini memastikan bahwa struktur *FP-Tree* yang dibentuk hanya terdiri dari item-item yang relevan dan sering muncul, sehingga proses pencarian *frequent pattern* dapat berjalan lebih cepat, terarah, dan menghasilkan pola yang lebih bermakna. Daftar item yang lolos seleksi ditampilkan pada Tabel 2.

Tabel 2. *Item Ordered*

Kode	<i>Sup. Count</i>
K7	45
K1	41
C5	38
C6	34

Kode	Sup. Count
S2	33
G1	30
D4	26
K3	25
B2	23
C3	23
M2	23
B3	21
C2	21
S7	20
B6	18
H2	16
R2	16
B4	15
P1	15

Setelah diketahui item-item yang memenuhi *minimum support* serta urutannya, langkah selanjutnya adalah mengurutkan item pada setiap transaksi berdasarkan nilai *support* dari yang terbesar hingga terkecil seperti pada Tabel 3 di bawah ini.

Tabel 3. Data Transaksi *Ordered*

<i>TID</i>	Daftar Barang
1	K7, K1, C5, C6, S2, B6
2	K1, S2, G1, B6
3	K7, K1, C6, G1, D4, K3, B3, B6, H2, R2
4	K7, K1, S2, G1, D4, C3, B3, B6, B4
5	K7, C5, G1, D4, B2, M2, C2, S7, B6, P1
6	K7, C5, S2, G1, B2, M2
7	S2, D4, M2, B3, B6, P1
8	K7, K1, M2, C2, S7, P1
9	K7, K1, C5, C6, D4, M2
10	K7, K1, C5, C6, G1, M2, R2, P1
11	K7, K1, C5, S2, M2, S7
12	K1, C5, C6, S2, D4, K3, R2, B4
13	K7, K1, C5, S2, G1, K3, C3, P1
14	K1, C6, S2, G1, K3, B3, C2, B6
15	K7, C5, C6, S2, K3, B2
16	C5, C6, D4, K3, B3, C2, H2

<i>TID</i>	Daftar Barang
17	K7, K1, C5, C6, S2, G1, D4, K3, B2, C3, R2
18	K7, K1, C5, G1, D4, K3, B2, M2, B3, B4
19	K7, C6, S2, B3, C2, B4
20	K1, C5, S2, S7, P1
21	K7, C6, S2, G1, S7
22	K7, C5, C6, B2, S7, B4, P1
23	K7, K1, C5, C6, S2, K3, B2, C3, C2
24	K1, C6, S2, M2, C2, B6
25	K7, K1, C5, C6, S2, K3, C3, B4
26	K1, S2, K3, M2, C2, S7, P1
27	K7, K1, C5, C6, S2, K3, C2
28	K7, K1, C5, S2, D4, K3, C3, R2
29	K7, K1, B3, S7, B6, R2, P1
30	K7, K1, C5, S2, G1, D4, B2, C3, B3, S7
31	K1, C5, C6, S2, D4, K3, B2, C3, B3
32	K7, K1, C5, C6, S2, K3, B2, C3, H2, B4
33	K7, G1, C3, M2, C2, B6, H2, B4, P1
34	K1, C5, C6, D4, K3, B2, M2, B3, S7, H2, R2
35	K7, C5, C6, K3, B2, B3, B6
36	K1, C6, G1, D4, B2, M2, H2, B4, P1
37	K7, S2, G1, M2, S7, H2, B4
38	K1, C6, M2, C2, S7, B4
39	K7, K1, S2, G1, K3, C3, H2, R2
40	K7, K1, C5, C3, B3, B6, R2
41	K7, K1, C6, K3, C3, M2, S7
42	K7, G1, D4, B2, B3, C2, S7, P1
43	C5, G1, D4, C3, C2, S7
44	K7, G1, C3, B3, S7, H2, R2
45	K7, K1, C5, S2, G1, C3, M2, C2, S7, B6, B4
46	K7, C6, K3, B2, C3, B3
47	K7, K1, C5, G1, K3, C2, B6, B4
48	K7, K1, C5, G1, D4, B2, H2, R2
49	K7, C5, C6, S2, M2, S7, H2
50	K7, C6, S2, D4, C3, H2, R2
51	K1, C5, S2, K3, C3, H2, B4
52	K7, C6, G1, D4, B3, C2, H2
53	K7, C5, G1, D4, K3, B2, M2, B3
54	K7, K1, C6, G1, D4, C3, B3

TID	Daftar Barang
55	K7, S2, D4, M2, S7, P1
56	K7, K1, C5, C6, S2, G1, B2
57	K1, C5, C6, K3, C3, C2, B6
58	K7, C6, S2, G1, B2, C3, B3, C2, B6, R2
59	K7, K1, C5, C6, M2, R2, P1
60	G1, D4, B2, C3, B3, H2, R2
61	K1, C5, G1, D4, B2, C2, B6, H2, R2, P1
62	K1, C5, C6, K3, M2, C2, B6
63	C5, G1, D4, B2, M2, C2, B4
64	K7, K1, C5, C6, S2, D4, B2, S7

3.4 Pembentukan FP-Tree

FP-Tree atau *Frequent Pattern Tree* adalah sebuah struktur pohon yang digunakan algoritma *FP-Growth* yang bertujuan untuk menemukan *frequent itemset* (pola item yang sering muncul). Untuk membentuk pohon ini, sistem akan memindai *dataset* dua kali, pemindaian pertama bertujuan menghitung *support count* setiap item. Selanjutnya pemindaian ke dua untuk menyaring setiap item yang memenuhi *support count* sehingga bisa diurutkan dan dibentuk menjadi *FP-Tree*. Proses ini dilakukan untuk menghindari pembangkitan kandidat secara eksplisit (langsung) seperti pada algoritma *Apriori* sehingga membuat proses lebih cepat.

Pada tahap pembentukan *FP-Tree*, hanya item yang telah lolos *minimum support* pada Tabel 2 yang digunakan. Setiap transaksi kemudian diurutkan berdasarkan nilai *support* tertinggi ke terendah agar struktur *tree* lebih kompak dan memudahkan pencarian pola. Setelah transaksi diurutkan, masing-masing transaksi dimasukkan ke dalam *FP-Tree* dengan menambahkan *node* baru atau memperbarui nilai hitungan (*count*) *node* yang sudah ada apabila *path* yang sama dilalui kembali.

Visualisasi proses pembentukan *FP-Tree* ditampilkan pada Gambar 2 dan Gambar 3 untuk menunjukkan bagaimana *tree* berkembang seiring masuknya transaksi.

1) *FP-Tree* ke-1



Gambar 2. *FP-Tree* ke-1

Pada Gambar 2, pembuatan *FP-Tree* pertama kali dilakukan dengan membuat *Root* terlebih dahulu, setelah itu *Root* akan dibuat cabang baru yang di isi dengan daftar transaksi yang sudah diurutkan. *FP-Tree* ke-1 ini terbentuk dari *TID* 1 yaitu: K7, K1, C5, C6, S2, dan B6. Bisa dilihat pada *FP-Tree* ke-1, *Root* tersebut mempunyai 1 cabang yaitu K7, pada cabang tersebut terdapat angka 1 yang berarti cabang masih dilalui oleh item tersebut 1 kali saja. Jika nanti cabang tersebut dilewati lagi maka nilai akan ditambahkan 1, dan begitu seterusnya hingga *FP-Tree* selesai terbentuk dari semua *TID*.

2) *FP-Tree* ke-2

Setelah *FP-Tree* ke-1 dari *TID* pertama terbentuk, maka selanjutnya akan ditambahkan *TID* kedua yang akan menjadi *FP-Tree* ke-2 seperti pada Gambar 3. Penambahan *TID* kedua ini juga dimulai dari *Root* terlebih dahulu, jika *Root* tidak punya cabang item tersebut, maka *Root* akan membuat cabang baru. *FP-Tree* ke-2 ini terbentuk dari *TID* pertama, dan *TID* kedua yaitu: K1, S2, G1, dan B6.



Gambar 3. *FP-Tree* ke-2

3.5 Menghitung CPB, CFT, dan FPG

CPB, CFT, dan FPG adalah perhitungan yang berguna untuk mencari *rule* atau jalur dari *FP-Tree* mana yang memenuhi syarat. *Conditional Pattern Base (CPB)* adalah rute yang didapat dari *FP-Tree*. Selanjutnya ada *Conditional FP-Tree (CFT)* yaitu berisi item apa saja yang memenuhi *minimum support* dari rute CPB tadi, Dan yang terakhir adalah *Frequent Pattern Generate (FPG)* yang berisi kombinasi final item-item yang memenuhi *minimum support*. Karena terlalu banyak item dan *rule* yang dihasilkan, maka peneliti akan menampilkan 4 item yang memenuhi *minimum support* seperti pada Tabel 4 berikut ini.

Tabel 4. Tabel CPB, CFT, FPG

Item	CPB	CFT	FPG
S2	{K7, K1, C5, C6 : 8}, {K1 : 2}, {K7, K1 : 2}, {K7, C5 : 1}, {K7, K1, C5 : 5}, {K1, C5, C6 : 2}, {K1, C6 : 2}, {K7, C5, C6 : 2}, {K7, C6 : 4}, {K1, C5 : 2}, {K7 : 2}	{K7 : 24}, {K1 : 23}, {C5 : 20}, {C6 : 18}, {K7, K1 : 15}	{K7, S2 : 24}, {K1, S2 : 23}, {C5, S2 : 20}, {C6, S2 : 18}, {K7, K1, S2 : 15}
C6	{K7, K1, C5 : 11}, {K7, K1 : 3}, {K7, C5 : 4}, {K7 : 6}, {K1, C5 : 5}, {K1 : 4}, {C5 : 1}	{K7 : 24}, {K1 : 23}, {C5 : 21}, {K7, C5 : 15}, {K1, C5 : 16}	{K7, C6 : 24}, {K1, C6 : 23}, {C5, C6 : 21}, {K7, C5, C6 : 15}, {K1, C5, C6 : 16}
C5	{K7, K1 : 20}, {K7 : 7}, {K1 : 8}	{K7 : 27}, {K1 : 28}, {K7, K1 : 20}	{K7, C5 : 27}, { K1, C5 : 28}, {K7, K1, C5 : 20}
K1	{K7 : 27}	{K7 : 27}	{K7, K1 : 27}

Pada Tabel 4, karena banyaknya hasil FPG yang ada dari beberapa item, maka peneliti memilih salah satu hasil dari perhitungan FPG untuk melanjutkan contoh perhitungan Mencari *strong rule*, yaitu pada baris C5 dengan *rule FPG* K7, K1, dan C5 yang memiliki nilai 20.

3.6 Menentukan Strong Rule

Perhitungan *strong rule* ini dilakukan untuk mengetahui apakah kombinasi item tersebut memiliki keterkaitan antar item, hal ini bisa dimanfaatkan untuk mengetahui aturan item. Untuk mengetahuinya kita harus mengetahui dulu *support* dari kombinasi item yang bersangkutan yaitu K7, K1, dan C5. Karena terdapat 3 item yang bersangkutan, peneliti hanya membutuhkan *support* dari 1 *itemset* dan 2 *itemset* seperti pada Tabel 5. Untuk mencari *support* item, hanya perlu menghitung berapa banyak munculnya kombinasi item yang ada pada *database*.

Tabel 5. Support 2 itemset

Itemset	support
K7, K1	27
K7, C5	27
K1, C5	27
K7	45
K1	41
C5	38

Setelah mengetahui *support* item, selanjutnya akan dilakukan perhitungan *strong rule*. Berikut adalah rumus yang dipakai:

$$Confidence (K7, K1 \rightarrow C5) = \frac{Support (K7, K1, C5)}{antecedent (K7, K1)}$$

Tabel 6. Perhitungan Strong Rule

Rule	Perhitungan	Conf
K7, K1 $\rightarrow$ C5	20/27	0,7407
K7, C5 $\rightarrow$ K1	20/27	0,7407
K1, C5 $\rightarrow$ K7	20/27	0,7407
K7 $\rightarrow$ K1, C5	20/45	0,4444
K1 $\rightarrow$ K7, C5	20/41	0,4878
C5 $\rightarrow$ K7, K1	20/38	0,5263

Pada Tabel 6, dari 6 *rule* yang dibuat, terdapat 3 *rule* yang tidak memenuhi *minimum confidence* 0,7 atau 70%. Dan terdapat 3 *rule* yang memenuhi *minimum confidence* yaitu dengan nilai 0,7407 atau bisa dikatakan 74,07%.

Berikut adalah penjabaran yang bisa diartikan dari *rule* yang lolos seleksi.

K7 = Kemeja

K1 = Kaos

C5 = Celana Panjang

- 1) K7, K1 $\rightarrow$ C5, bisa diartikan bahwa jika pembeli membeli kemeja dan kaos, maka kemungkinan pembeli juga akan membeli celana panjang adalah 74%.
- 2) K7, C5 $\rightarrow$ K1, bisa diartikan bahwa jika pembeli membeli kemeja dan celana panjang, maka kemungkinan pembeli juga akan membeli kaos adalah 74%.
- 3) K1, C5 $\rightarrow$ K7, bisa diartikan bahwa jika pembeli membeli kaos dan celana panjang, maka kemungkinan pembeli juga akan membeli kemeja adalah 74%.

Dari perhitungan *strong rule* ini, kombinasi *rule* item di atas adalah kombinasi yang kuat untuk dibeli secara bersamaan, sehingga pemilik toko bisa membuat strategi pemasaran berdasarkan hasil tersebut. Contohnya bisa dibuat *bundle* celana panjang dan kemeja, bisa juga dengan diletakkan secara berdekatan agar konsumen melakukan pembelian yang impulsif, atau juga dapat dimanfaatkan untuk *cross selling*, sehingga dapat berpotensi meningkatkan penjualan pada toko.

3.7 Pengujian Lift Ratio

Setelah menemukan *rule* yang kuat, untuk memastikan keakuratan hubungan antar item maka bisa dilakukan pengujian *lift ratio*, pengujian ini bertujuan menilai apakah hubungan antar item ini benar-benar memiliki hubungan, atau hanya kebetulan karena mereka sama-sama sering muncul. Berikut adalah rumus yang digunakan untuk menghitung nilai *lift ratio*.

$$Lift\ Ratio(K7, K1 \rightarrow C5) = \frac{Confidence\ (K7, K1 \rightarrow C5)}{Support\ (C5)}$$

Tabel 7. Perhitungan Lift Ratio

Rule	Support Antecedent	Conf.	Lift Ratio
K7, K1 → C5	0,5938	0,7407	1,2472
K7, C5 → K1	0,6406	0,7407	1,1563
K1, C5 → K7	0,7031	0,7407	1,0526

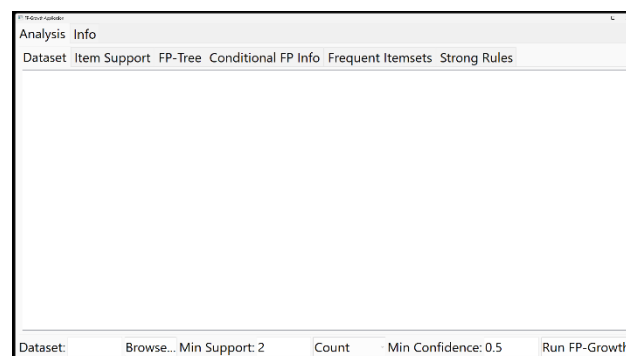
Dapat dilihat pada Tabel 7, bahwa semua *rule* memenuhi aturan *lift ratio*, hal ini dikarenakan semua *rule* mendapat nilai *lift ratio* > 1 yang artinya semua kombinasi item dari K7, K1, dan C5 saling memiliki keterkaitan yang kuat. Perlu diketahui, nilai *lift ratio* bukanlah nilai kepastian, tapi adalah indikasi *probabilistik* (peluang). Nilai *lift* > 1 menunjukkan bahwa kemunculan *consequent* lebih mungkin terjadi bersama *antecedent* dibandingkan secara acak. Namun nilai *lift* tidak menjamin kepastian, melainkan menunjukkan kecenderungan atau potensi hubungan antar item.

Berdasarkan hasil pengujian pada Tabel 7, aturan dengan nilai *lift ratio* tertinggi adalah K7, K1 → C5 dengan nilai 1,2472, yang menunjukkan bahwa pelanggan yang membeli K7 dan K1 memiliki kecenderungan 1,24 kali lebih besar untuk membeli C5. Ini menjadi pola asosiasi terkuat dari seluruh kombinasi yang diuji dan menjadi temuan utama penelitian. Pola ini dapat dijadikan dasar strategi pemasaran, seperti penempatan produk secara berdampingan atau rekomendasi produk, dan strategi pemasaran lainnya.

3.8 Implementasi Algoritma ke Aplikasi

Setelah mengetahui bagaimana cara perhitungan algoritma *FP-Growth*, untuk mendukung proses analisis pola pembelian, peneliti mengembangkan sebuah aplikasi

sederhana yang mengimplementasikan algoritma *FP-Growth*. Aplikasi ini berfungsi untuk memvisualisasikan setiap tahap perhitungan, sehingga pengguna dapat memahami alur algoritma secara lebih intuitif.



Gambar 4. Tampilan Awal Aplikasi

Pada halaman awal, aplikasi dibuka akan menampilkan tampilan kosong seperti pada Gambar 4, untuk menjalankannya, pengguna aplikasi harus memiliki *dataset* yang siap pakai, yang nantinya akan di unggah pada aplikasi tersebut.

Setelah *dataset* diunggah, maka aplikasi akan menampilkan isi dari *dataset* tersebut dalam bentuk tabel seperti pada Gambar 5. Untuk menjalankan perhitungan *FP-Growth*, pengguna bisa memasukkan nilai *minimum support* dan *minimum confidence* pada tempat yang sudah disediakan. Setelah itu pengguna bisa menekan tombol *Run FP-Growth* dan tunggu hingga notifikasi perhitungan muncul.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	C6	K7	S2	P2	S3	C5	T5	B6	K1	T4	S5						
2	T6	S2	S4	K8	K1	J3	S3	T3	B6	G1	P5						
3	B3	C6	H1	K1	R2	K3	K7	D4	B6	H2	G1						
4	B4	S2	K2	K1	B3	C3	D4	K7	B6	T3	S4	G1	C4				
5	K7	M1	C5	B6	C4	D4	B2	G1	T6	C2	S7	M2	P1				
6	K7	B2	G1	M2	C5	S2	T4	S8	P3	J2							
7	D4	M5	B6	C22	B3	S2	B1	T3	M2	P1	C9						
8	K7	K1	C2	S6	M3	P5	C8	M2	J3	S7	P1						
9	M2	G2	M1	C6	S5	D4	K1	C5	K7	S3							
10	K1	C1	K7	R2	C6	C5	G1	M2	J3	P1							
11	K1	C5	K7	K4	S2	S7	D3	M2	S6	A1	J3	D2					
12	S2	M4	D4	K3	T6	K1	C6	B4	C22	R2	C5						

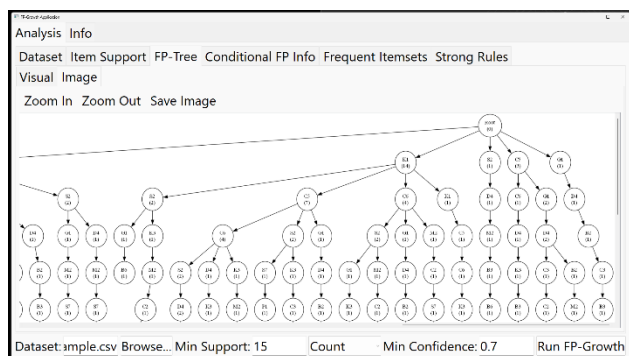
Gambar 5. Tampilan Isi Dataset

Pengguna bisa melihat perhitungan algoritma *FP-Growth* dengan memilih tab-tab yang ada di atas, tab-tab tersebut berisi langkah-langkah dari perhitungan algoritma yang dipakai. Dari menampilkan *support* item, *FP-Tree*, perhitungan *CPB*, *CFT*, *FPG*, sampai dengan menemukan *strong rules*. Sehingga pengguna bisa menganalisis data dengan mudah, cepat, dan efisien.

Item	Support Count
1 K7	45
2 K1	42
3 C5	39
4 C6	35
5 S2	33
6 G1	30
7 D4	26
8 K3	25
9 B2	23
10 C3	23
11 M2	23
12 B3	21

Gambar 6. Tampilan *Support* item

Pada Gambar 6 terdapat halaman *support item*, menu ini menampilkan hasil perhitungan *support* untuk setiap item. Tahap ini secara otomatis menghitung jumlah kemunculan setiap item pada *dataset*, kemudian menyaring item-item yang tidak memenuhi *minimum support*. Hasil pada halaman ini merupakan dasar untuk menentukan *frequent 1-itemset*.

Gambar 7. Tampilan *FP-Tree*

Halaman *FP-Tree* pada Gambar 7 akan menampilkan struktur pohon yang dibentuk berdasarkan transaksi yang telah diurutkan sesuai frekuensi item. Proses pembentukan pohon dilakukan secara bertahap, dimulai dari *Root*, kemudian setiap transaksi ditambahkan sambil memperbarui nilai *count node*. Visualisasi ini membantu pengguna memahami bagaimana transaksi yang sama membentuk jalur yang berulang dalam *FP-Tree*.

CPB	CFT	FPG
1 {C5}	29	0.7436
2 {C5, K1}	28	0.7179
3 {C5, K7}	25	0.7143
4 {C6}	24	0.7273
5 {S2}	23	0.7667
6 {G1}	23	0.8400
7 {D4}	21	0.7143
8 {K3}	20	0.7407
9 {B2}	19	0.8261
10 {C3}	19	0.7600
11 {M2}	18	0.7826
12 {B3}	17	0.7391

Gambar 8. Tampilan *CPB, CFT, FPG*

Pada Gambar 8, terdapat tampilan dari halaman yang menampilkan tiga tahapan utama pada proses pencarian

frequent pattern, yaitu *CPB (Conditional Pattern Base)* yang menunjukkan jalur-jalur transaksi yang berkaitan dengan suatu item tertentu, *CFT (Conditional FP-Tree)* yang menampilkan pohon turunan yang terbentuk dari *CPB* setelah item yang tidak memenuhi *support* dihilangkan, dan *FPG (Frequent Pattern Generate)* yang berisi hasil akhir berupa kombinasi item yang memenuhi *minimum support*. Tampilan ini bisa mempermudah pengguna dalam menelusuri pola tanpa harus membaca *FP-Tree* secara manual.

Itemset	Support
1 {C5}, {K1}	29
2 {C5}, {K7}	28
3 {K7}, {K1}	27
4 {K7}, {C6}	25
5 {K1}, {C6}	24
6 {K7}, {S2}	24
7 {G1}, {K7}	23
8 {S2}, {K1}	23
9 {C5}, {C6}	22
10 {K3}, {K1}	21
11 {C5}, {S2}	20
12 {C5}, {K7}, {K1}	20

Gambar 9. Tampilan *Frequent Itemset*

Halaman *frequent itemset* pada Gambar 9 berisi kumpulan dari seluruh kombinasi item dari *database* yang memenuhi *minimum support* lengkap dengan nilai *support* dari *itemset*. Dari halaman ini, pengguna bisa mendapatkan Informasi untuk mengidentifikasi pola pembeli yang sering muncul secara bersamaan.

Rule	Support	Confidence
1 {C5} -> {K1}	29	0.7436
2 {C5} -> {K7}	28	0.7179
3 {C6} -> {K7}	25	0.7143
4 {S2} -> {K7}	24	0.7273
5 {G1} -> {K7}	23	0.7667
6 {K3} -> {K1}	21	0.8400
7 {C5, K7} -> {K1}	20	0.7143
8 {K7, K1} -> {C5}	20	0.7407
9 {B2} -> {C5}	19	0.8261
10 {K3} -> {C5}	19	0.7600
11 {C3} -> {K7}	18	0.7826
12 {B2} -> {K7}	17	0.7391

Gambar 10. Tampilan *Strong Rules*

Tahap terakhir adalah pembentukan aturan asosiasi menggunakan nilai *confidence*. Pada tahap ini, aplikasi akan menghitung *confidence* untuk setiap *rule* yang memungkinkan. Dan setiap *rule* yang memenuhi *minimum confidence* akan ditampilkan pada halaman ini. Dengan halaman ini pengguna dapat mengetahui *rule* mana yang memiliki nilai *support* dan *confidence* yang kuat.

4) KESIMPULAN

Berdasarkan data dan hasil penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

- 1) Penerapan algoritma *FP-Growth* pada data histori penjualan di toko pakaian TREND berhasil menemukan pola asosiasi antar produk yang sering

dibeli secara bersamaan. Pola-pola tersebut divisualisasikan melalui *FP-Tree* dan menghasilkan *frequent itemsets*, serta aturan asosiasi yang valid berdasarkan nilai *support* dan *confidence*. Temuan aturan ini dapat digunakan untuk mendukung strategi pemasaran, seperti penempatan produk yang berdekatan untuk pembelian yang impulsif, promosi *bundling*, *cross selling* dan strategi pemasaran lainnya sehingga dapat berpotensi meningkatkan penjualan pada toko.

- 2) Hasil pengujian nilai *lift ratio* menunjukkan bahwa kombinasi item K7 (kemeja), K1 (kaos), dan C5 (celana panjang) merupakan pola pembelian terkuat dalam penelitian ini. Aturan asosiasi yang dihasilkan memiliki nilai *lift ratio* 1,2472 dan 1,1563, yang menandakan adanya korelasi positif antar item. Hal ini menunjukkan bahwa pola pembelian yang ditemukan oleh algoritma *FP-Growth* tidak terjadi secara kebetulan, tetapi memiliki hubungan yang signifikan sehingga dapat dijadikan dasar yang kuat dalam penyusunan strategi pemasaran yang efektif dan tepat sasaran.

Secara keseluruhan proses perhitungan algoritma *FP-Growth* telah memberikan hasil yang baik dan relevan bagi peneliti. Namun, perlu diakui bahwa kualitas pola yang ditemukan sangat dipengaruhi oleh jumlah dan kelengkapan data. Pada penelitian ini, jumlah sampel hanya 64 transaksi dan masih terbatas pada kategori barang, sehingga hasil yang diperoleh belum merepresentasikan keseluruhan data penjualan toko yang sebenarnya berjumlah ribuan. Dengan *dataset* yang lebih besar dan lebih detail, pola yang ditemukan akan jauh lebih kaya dan dapat menghasilkan strategi pemasaran yang lebih baik.

Selain itu, aplikasi *FP-Growth* yang dikembangkan dalam penelitian ini masih bersifat sederhana karena keterbatasan waktu dan pengalaman peneliti yang kurang. Oleh karena itu, pengembangan lebih lanjut sangat disarankan, baik dari sisi tampilan, performa, maupun penambahan fitur seperti analisis visual pola, rekomendasi otomatis, dan integrasi dengan data penjualan *real-time* agar pengguna dapat memanfaatkan aplikasi secara lebih optimal, aman, dan nyaman.

Ucapan Terima Kasih

Penulis menyampaikan rasa terima kasih kepada seluruh pihak di Fakultas Teknik dan Informatika Universitas Islam Balitar, dosen-dosen yang berkaitan dengan penelitian ini, dan juga teman-teman sesama program studi karena telah memberikan dukungan, masukan, dan motivasi sehingga penulis dapat menyelesaikan penelitian sampai mempublikasikan hasil penelitian ini.

DAFTAR PUSTAKA

- [1] D. Kurniawan, M. Sahata Sipayung, R. Ismayanti, M. Rivani Ibrahim, Y. Bintan, and S. Aulia Miranda, "Optimalisasi Strategi Pemenuhan Persediaan Stok Barang Menggunakan Algoritma Frequent Pattern Growth," *METIK JURNAL*, vol. 6, no. 2, pp. 104–114, Dec. 2022, doi: 10.47002/metik.v6i2.387.
- [2] M. I. Al Machmudi, "Gerak Cepat Bisnis Ritel Beradaptasi pada Perubahan Tren dan Teknologi." Accessed: Feb. 17, 2025. [Online]. Available: https://mediaindonesia.com/ekonomi/743663/gerak-kecepatan-bisnis-ritel-beradaptasi-pada-perubahan-tren-dan-teknologi?form=MG0AV3#google_vignette
- [3] Kharomiyah, N. Rahaningsih, and R. D. Dana, "Analisis Keterkaitan Penjualan Obat melalui Penerapan Algoritma FP-Growth guna Optimalisasi Strategi Pemasaran," *Jurnal SAINTIKOM (Jurnal Manajemen Informatika dan Komputer)*, vol. 23, pp. 57–67, Feb. 2024, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jis/index>
- [4] Firmansyah and A. Yulianto, "Market Basket Analysis for Books Sales Promotion using FP Growth Algorithm, Case Study: Gramedia Matraman Jakarta," *Journal of Informatics and Telecommunication Engineering*, vol. 4, no. 2, pp. 383–392, Jan. 2021, doi: 10.31289/jite.v4i2.4539.
- [5] Muttaqin et al., *Pengenalan Data Mining*. Yayasan Kita Menulis, 2023. Accessed: Dec. 22, 2024. [Online]. Available: <https://repository.upy.ac.id/4946/1/FullBook%20Pengenalan%20Data%20Mining.pdf>
- [6] F. Suliantana, *Buku Dasar Data Mining from A to Z*, 2024. Accessed: Dec. 22, 2024. [Online]. Available: https://www.researchgate.net/publication/377018853_Buku_Dasar_Data_Mining_from_A_to_Z_-_Feri_SLN_Free
- [7] Mustika et al., *Data Mining dan Aplikasinya*. Widina Bhakti Persada Bandung (Grup CV. Widina Media Utama), 2021. Accessed: Dec. 22, 2024. [Online]. Available: <https://repository.penerbitwidina.com/media/publications/351768-data-mining-dan-aplikasinya-74a855a3.pdf>
- [8] P. W. Rahayu et al., *Buku Ajar Data Mining*. PT. Sonpedia Publishing Indonesia, 2024. Accessed: Dec. 22, 2024. [Online]. Available: https://www.researchgate.net/publication/377415198_BUKU_AJAR_DATA_MINING
- [9] M. Huda, *Algoritma Data Mining: Analisis Data Dengan Komputer*. 2019.

- [10] A. N. S. Putro and R. I. Gunawan, "Implementasi Algoritma FP-Growth untuk Strategi Pemasaran Ritel Hidroponik (Studi Kasus: PT. HAB)," *Jurnal Buana Informatika*, vol. 10, pp. 11–18, Apr. 2019, Accessed: Dec. 22, 2024. [Online]. Available: <https://ojs.uajy.ac.id/index.php/jbi/article/view/1746>
- [11] Dicoding Intern, "Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya," 2023. Available: <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>
- [12] A. Ma'arif, *Buku Ajar Pemrograman Lanjut Bahasa Pemrograman Python*, 2020.
- [13] B. Aprellia, S. Lestanti, and S. N. Budiman, "Penerapan Data Mining Menggunakan Algoritma FP-Growth Untuk Menentukan Pola Pembelian Produk Pada Minimarket Justin Mart," Aug. 2023.
- [14] R. Fauzi, A. W. Aranski, N. Nopriadi, and E. Hutabri, "Implementasi Data Mining pada Penjualan Pakaian dengan Algoritma FP-Growth," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, p. 436, Apr. 2023, doi: 10.30865/jurikom.v10i2.5795.
- [15] M. Rizky, A. A. Ridha, and K. Prihandani, "Penentuan Paket Promosi Pakaian PT. D&C Production dengan Menggunakan Algoritma FP-Growth," vol. 5, no. 2, pp. 177–186, Dec. 2021, doi: 10.29408/edumatic.v5i2.3714.