

IMPLEMENTASI ALGORITMA FP-GROWTH MENGGUNAKAN ASSOCIATION RULE PADA MARKET BASKET ANALYSIS

Fitriyani

Fakultas Teknik, Universitas BSI Bandung
Jalan Sekolah Internasional No. 1-6, Bandung 40282, Indonesia
fitriyani.fyn@bsi.ac.id

Abstract - The set of data can be processed into information or useful knowledge, one of the data that can be processed is data purchases by consumers. However, large data processing will take a long time in the process. So that these data require appropriate methods in the process. The method is often used in data processing transactions are Apriori, but a great deal less precise data using Apriori because in the process repeatedly scanning the database (candidate set generation). In this study using the FP-Growth method for determining frequent itemset with structure of FP-Tree and Association Rule to determine support and confidence in the transaction data so that the results can be known relationships between an item with other items that are frequently purchased by consumers.

Keywords : Apriori, FP-Growth, Association Rule, Transaction, Frequent Itemset.

Abstrak - Himpunan data yang besar dapat diolah menjadi informasi atau pengetahuan yang bermanfaat, salah satu data yang dapat diolah adalah data transaksi pembelian barang oleh konsumen. Akan tetapi pemrosesan data yang besar akan membutuhkan waktu yang lama dalam prosesnya. Sehingga data tersebut membutuhkan metode yang tepat dalam proses pengolahannya. Metode yang sering digunakan dalam pengolahan data transaksi adalah Apriori, akan tetapi data yang besar kurang tepat menggunakan Apriori karena dalam prosesnya melakukan *scanning* berulang kali pada database (*candidate set generation*). Dalam penelitian ini menggunakan metode FP-Growth untuk menentukan *frequent itemset* dengan struktur FP-Tree dan *Association Rule* untuk menentukan *support* dan *confidence* pada data transaksi sehingga hasilnya dapat diketahui hubungan-hubungan antara suatu barang dengan barang lainnya yang sering dibeli oleh konsumen.

Kata Kunci : Apriori, FP-Growth, Association Rule, Transaction, Frequent Itemset.

PENDAHULUAN

Database penjualan mempunyai data transaksi yang sangat besar jika setiap harinya ada 100 transaksi, dalam sebulan ada sekitar 3.000 transaksi dan setahun menjadi sekitar 36.000 transaksi, kemudian dalam waktu dua tahun data tersebut sudah menjadi 72.000. Data transaksi atau *database* penjualan yang sangat besar, apakah dimanfaatkan lagi atau hanya disimpan sebagai arsip? Jika hanya disimpan

tanpa digunakan lagi data ini akan terus menumpuk sehingga untuk memelihara data tersebut membutuhkan biaya pemeliharaan dan ini akan masuk kedalam biaya tambahan perusahaan sehingga perusahaan harus menyediakan biaya untuk pemeliharaan data tersebut.

Himpunan data penjualan yang dimiliki sebenarnya dapat diolah menggunakan data mining untuk melihat pola pembelian pelanggan, dengan data mining untuk data yang besar tidak akan terbuang begitu saja dan dapat

bermanfaat sehingga dapat memberikan keuntungan kepada perusahaan. Data mining dasarnya adalah untuk membangun program komputer yang menyaring *database* otomatis, mencari keteraturan atau pola. Pola yang kuat dan jika ditemukan adanya kemungkinan-kemungkinan akan menggeneralisasi untuk membuat prediksi yang akurat pada data masa depan. Akan tetapi akan ada masalah ketika banyak pola yang tidak menarik (Witten, dkk: 2011). Data mining sebagai proses untuk mendapatkan informasi yang berguna dari gudang basis data yang besar dan dapat diartikan sebagai pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Istilah data mining kadang disebut juga *knowledge discovery* (Prasetyo: 2012). Mesin pencari google memanfaatkan data mining dalam mengolah data permintaan informasi untuk para browser. Seperti contoh orang yang sedang sakit Flu ketika mencari informasi tentang penyakit ini, itu akan membedakan pencarian informasi pada orang yang hanya ingin mengetahui apa yang dimaksud dengan penyakit Flu. Dari sekian banyak permintaan informasi mengenai apapun, Google memanfaatkan data tersebut untuk pengetahuan yang bermanfaat, sehingga dapat memenuhi tantangan global saat ini.

Perusahaan retail seperti supermarket mempunyai data yang begitu besar meliputi data penjualan, data penjualan yang berisi ratusan bahkan ribuan data transaksi dapat diolah dengan data mining sehingga data tersebut tidak hanya menjadi data arsip, tetapi dapat memberikan pengetahuan tentang pola pikir konsumen, seperti tipe-tipe pembeli, hubungan antar item yang dibeli. Seperti tipe pembeli ibu rumah tangga ketika belanja kebutuhan, sebagian besar mereka membeli bahan pokok. Tetapi berbeda untuk tipe pembeli pelajar, ketika masuk tahun ajaran baru mereka akan membeli kebutuhan untuk belajar, dan sebagainya. Kesimpulannya adalah market basket merupakan pengolahan data keranjang belanja konsumen.

Contoh ibu rumah tangga sering membeli Gula kemudian Kopi, jarang sekali setelah membeli Gula akan membeli Kemeja dan Pulpen atau contoh lain yang paling umum di toko retail besar yang ada di US, Wal-Mart adalah pembelian Beer dan Diapers yang saling berhubungan, beer adalah minuman beralkohol dan diapers adalah merk popok bayi (Aprilla, dkk: 2013).

Dalam observasi diatas diketahui bahwa setiap hari jumat malam sekitar jam enam dan tujuh sore, laki-laki dewasa akan membeli beer dan popok secara bersamaan, para istri mereka akan meminta suaminya untuk membeli popok sedangkan para suami akan menghabiskan waktu hari minggu dengan meminum beer. Dari observasi tersebut memberikan keputusan pada pihak manajemen untuk display produk beer dan bersebelahan dengan produk diapers. Pola yang sudah terbentuk pada keranjang belanja atau market basket dapat memberikan keputusan untuk display produk, promosi diskon produk, strategi penjualan untuk barang yang kurang menarik minat beli konsumen dan sebagainya, pola ini disebut dengan analisis asosiasi (*analysis association*).

Data keranjang belanja dapat diubah menjadi biner untuk pengolahan data tersebut, dimana angka biner adalah berisi 1 (ya atau true) dan 0 (tidak atau false), pada data keranjang belanja untuk pembelian sebuah itemset X akan diberi angka 1, sedangkan tidak membeli itemset X akan diberi angka 0. *Association rule* adalah pernyataan implikasi $X \rightarrow Y$, dimana X dan Y merupakan itemset yang lepas (disjoint), sehingga memenuhi persyaratan $X \cap Y = \{ \}$. Kekuatan aturan asosiasi dapat diukur dengan *support* dan *confidence*. *Support* digunakan untuk menentukan seberapa banyak aturan dapat diterapkan pada set data, sedangkan *confidence* digunakan untuk menentukan seberapa sering item di dalam Y muncul dalam transaksi yang berisi X (Prasetyo: 2012).

Rumus untuk mencari *support* dan *confidence* :

$$\text{Support } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \dots(1)$$

$$\text{Confidence } (X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \dots(2)$$

Contoh jika diketahui $X = \{\text{Gula, Kopi}\}$ dan $Y = \{\text{Aqua}\}$, jumlah transaksi yang berisi $X \cup Y = \{\text{Gula, Kopi, Aqua}\}$ ada 1 dan total jumlah transaksi N ada 12 sehingga *support* untuk aturan $X \rightarrow Y$ menjadi $(X \rightarrow Y) = 1/12 = 0.083$. Sedangkan untuk mencari aturan *confidence* adalah dengan membagi jumlah $X \cup Y = \{\text{Gula, Kopi, Aqua}\}$ dengan jumlah *support* untuk $X = \{\text{Gula, Kopi}\}$, $\sigma(X \cup Y) = 1$ dan $\sigma(X) = 5$ sehingga *confidence* $(X \rightarrow Y) = 1/5 = 0.2$.

Hasil *support* dan *confidence* pada data mining untuk market basket dapat dibatasi dengan diterapkan *association rule*, dimana *minsup* dan *minconf* dapat diatur sesuai dengan kebutuhan. *Minsup* adalah ambang batas *support*, sedangkan *minconf* adalah ambang batas *confidence*. Sebagai contoh *minsup* 50% dan *minconf* 50% sehingga hasil dari pengolahan data yang akan tampil yang hanya mempunyai nilai *support* dan *confidence* 50% atau 0.5, data yang mempunyai nilai dibawah 50% atau 0.5 akan dibuang.

Algoritma dalam penelitian ini menggunakan algoritma FP-Growth yang merupakan salah satu algoritma untuk menentukan himpunan data yang sering muncul pada keranjang belanja konsumen atau dapat disebut dengan *frequent itemset*. Algoritma lainnya yang sering digunakan untuk menentukan *frequent itemset* adalah algoritma Apriori, paradigma Apriori yang dikembangkan oleh Agrawal dan Srikan (1994) adalah anti-monotone Apriori Heuristic dimana jika *itemset* pada panjang pola k *infrequent* (tidak sering muncul) dalam set data maka semua subset (bagian) dari *itemset* pasti juga

infrequent sehingga kandidat yang diperiksa menjadi berkurang. Dasar algoritma Apriori adalah mencari himpunan kandidat dengan panjang $(k+1)$ dari semua pola *frequent* dengan panjang k , kemudian kandidat yang memenuhi akan dicocokkan jumlah kemunculan pola tersebut dengan yang ada dalam database. Sehingga algoritma Apriori akan melakukan *scanning* database berulang-ulang, untuk data relatif kecil apriori dapat diterapkan untuk menentukan *frequent itemset*, akan tetapi untuk data yang besar algoritma Apriori akan membutuhkan waktu yang lama dalam menemukan pola (Samuel: 2008). Berbeda dengan algoritma FP-Growth yang hanya memerlukan dua kali *scanning* database dalam menghasilkan *frequent itemset*.

Algoritma FP-Growth adalah sebuah metode dalam data mining untuk mencari *frequent itemset* tanpa menggunakan *candidate generation*. Pembangunan data menggunakan struktur FP-Tree untuk mengolah database transaksi (Jiawei, dkk: 2012).FP-Growth menggunakan strategi *divide-conquer* sehingga algoritma ini hanya membutuhkan dua kali *scanning database*. Contoh $I = \{i_1, i_2, i_3, \dots, i_n\}$ adalah sekumpulan dari data item dan basis data transaksi $T = \{t_1, t_2, t_3, \dots, t_n\}$, setiap transaksi t_i berisi bagian dari item I . *Scanning* pertama dilakukan ketika *support count* (jumlah transaksi yang mengandung *itemset* tertentu) sudah ditentukan diawal, contohnya t_1 memiliki item i_1, i_2 kemudian tentukan *support count* $^{\xi} = 2$. Untuk menentukan *support* adalah berdasarkan kebutuhan karena *support* ini akan berpengaruh pada hasil FP-Tree yang dibuat dan hasil *support* dan *confidence* pada *association rule*. Semakin *support count* $^{\xi}$ bernilai tinggi 100% atau 10 maka akan sedikit node pada FP-Tree dan *support-confidence* akan sedikit menampilkan hasil pola hubungan antara item.

Kekurangan dalam algoritma Apriori diperbaiki oleh algoritma FP-Growth dengan menghilangkan *candidate generation*, karena algoritma FP-Growth menggunakan konsep pembangunan tree dalam

pencarian frequent itemset. Hal tersebut yang membuat algoritma FP-Growth lebih cepat daripada algoritma Apriori dalam pencarian *frequent itemset*

Algoritma FP-Growth menggunakan struktur data tree yang disebut dengan FP-Tree. Dalam penggalian itemset yang frequent menggunakan tiga langkah sebagai berikut :

1. Tahap pembangkitan *Conditional Pattern Base*, tahap ini merupakan subdatabse yang berisi prefix path dan *suffix pattern* (pola akhiran).
2. Tahap pembangkitan *Conditional FP-Tree*, pada tahap ini *support count* dari setiap item untuk *conditional pattern base* dijumlahkan.
3. Tahap pencarian *Frequent Itemset*, tahap ini merupakan lintasan tunggal (*single path*), kemudian didapatkan *frequent itemset* (Samuel: 2008).

dengan melakukan kombinasi item untuk conditional FP-Tree.

METODE PENELITIAN

Penelitian ini menggunakan dataset transaksi atau market basket dengan jumlah 12 transaksi dengan berbagai macam item yang dibeli oleh konsumen. Dataset tersebut akan diolah untuk mencari frequent itemset menggunakan metode algoritma FP- Growth dengan struktur pembangunan FP-Tree, kemudian dicari pola-pola belanja konsumen dengan menggunakan *association rule* untuk mencari *support* dan *confidence* pada item yang saling berhubungan.

PEMBAHASAN

Pembahasan dalam penelitian ini adalah dengan menerapkan FP-Growth pada data transaksi dan menggunakan *association rule*. Untuk *minsup* dan *minconf* diatur menjadi 50% atau 0.5 sehingga data *support* dan *confidence* yang tampil hanya yang memenuhi persyaratan tersebut.

FP-Tree

Dalam penelitian ini menggunakan sample data transaksi sebanyak 12 transaksi dengan berbagai macam item yang dibeli oleh konsumen. Adapun data tersebut sebagai berikut :

Tabel 1
Sampel Data Transaksi

TID	Priority Item
1	Gula, Kopi, Sampo, Sabun
2	Gula, Kopi, Sprei, Popok, Celana, Boneka, Kemeja
3	Sprei, Popok, Boneka
4	Gula, Aqua
5	Aqua, Sampo, Popok
6	Gula, Kopi, Sabun
7	Sprei, Celana, Boneka
8	Aqua, Sprei, Sampo, Popok, Sabun, Kemeja
9	Gula, Kopi, Celana
10	Aqua, Sampo
11	Gula, Kopi, Aqua
12	Sprei, Sampo, Sabun

Tentukan frekuensi kemunculan setiap item yang ada pada transaksi dan berikan prioritas untuk kemunculan item yang paling tinggi dengan rincian sebagai berikut :

Tabel 2
Frekuensi dan prioritas pada tiap item

Item	Frequent	Priority
Gula	6	1
Kopi	5	2
Aqua	5	3
Popok	4	6
Sprei	5	4
Sabun	4	7
Sampo	5	5
Kemeja	2	10
Celana	3	8
Boneka	3	9

Setelah dilakukan *scanning* data untuk menentukan *frekuensi* dan prioritas didapat bahwa item produk Gula memiliki nilai *frequent* paling tinggi dengan urutan prioritas pertama, kemudian Kopi urutan kedua, Aqua urutan ketiga, Sprei urutan keempat, Sampo urutan kelima, Popok urutan keenam, Sabun urutan ketujuh, Celana urutan kedelapan, Boneka urutan kesembilan dan terakhir Kemeja urutan kesepuluh. Langkah selanjutnya adalah menyesuaikan urutan pada tiap transaksi sesuai dengan *frequent* paling tinggi. Seperti pada tabel dibawah ini :

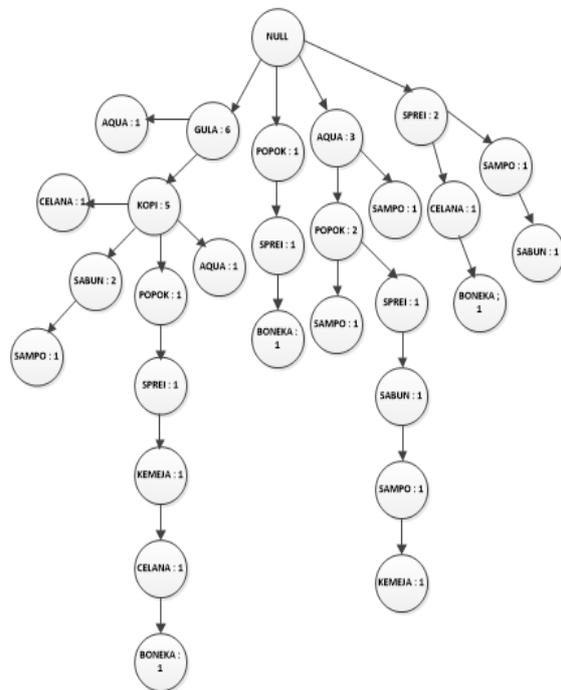
Tabel 3
Item Yang Sudah Diurutkan
Berdasarkan Prioritas

TID	Priority Item
1	Gula, Kopi, Sampo, Sabun
2	Gula, Kopi, Sprei, Popok, Celana, Boneka, Kemeja
3	Sprei, Popok, Boneka
4	Gula, Aqua
5	Aqua, Sampo, Popok
6	Gula, Kopi, Sabun
7	Sprei, Celana, Boneka
8	Aqua, Sprei, Sampo, Popok, Sabun, Kemeja
9	Gula, Kopi, Celana
10	Aqua, Sampo
11	Gula, Kopi, Aqua
12	Sprei, Sampo, Sabun

Pada transaksi pertama item yang dibeli dengan urutan Gula, Kopi, Sabun, Sampo terlihat pada tabel 1, sedangkan setelah dilakukan *scanning* pertama dan diurutkan berdasarkan *frequent* tertinggi dapat diurutkan untuk transaksi pertama menjadi Gula, Kopi, Sampo, Sabun karena produk Sampo lebih tinggi *frequentnya* dibandingkan dengan produk Sabun. Terlihat pada tabel 3, untuk transaksi kedua dengan urutan Gula, Kopi, Popok, Sprei, Kemeja, Celana, Boneka menjadi Gula, Kopi, Sprei,

Popok, Celana, Boneka, Kemeja karena produk Sprei lebih tinggi *frequentnya* dibandingkan dengan produk Popok, kemudian produk Celana lebih tinggi nilai *frequentnya* daripada produk Kemeja dan produk Boneka lebih tinggi nilai *frequentnya* daripada produk Celana serta produk Kemeja lebih tinggi nilai *frequentnya* daripada produk Boneka dan seterusnya.

Gambar dibawah ini memberikan ilustrasi mengenai pembentukan FP-Tree untuk data transaksi pada tabel 1.



Gambar 1
Ilustrasi Pembentukan FP-Tree

Ilustrasi pembentukan FP-Tree dapat dijelaskan sebagai berikut :

1. Pada gambar diatas adalah *scanning* kedua kemudian pembentukan FP-Tree setelah pengurutan item pada transaksi pertama, pembentukan FP-Tree dimulai dari node pertama diberi label Null selanjutnya node kedua diberi label Gula dan support count = 1 karena transaksi yang melaluilintasannya berjumlah 1. Node selanjutnya adalah Kopi dengan *support count* 1, Kopi berada setelah node Gula, kemudian

- Sabun berada setelah node Kopi dan Sampo berada setelah node Sabun.
2. Setelah lintasan pertama selesai, digambarkan lintasan kedua dengan urutan Gula dan Kopi dengan *support count* = 2, kemudian setelah Kopi ada node baru dengan label Popok, Sprei, Kemeja, Celana, Boneka.
 3. Lintasan kedua selesai, dilanjutkan dengan lintasan ketiga. Berbeda dengan lintasan pertama dan kedua, lintasan ketiga diawali dengan Popok dimana Popok merupakan node baru sehingga mempunyai urutan Null → Popok → Sprei → Boneka.
 4. Selanjutnya lintasan keempat merupakan node yang sudah ada yaitu Gula sehingga *support count* dari Gula menjadi 3, kemudian didalam node Gula ada node Aqua dengan *support count* 1.
 5. Proses selanjutnya adalah lintasan kelima yang diawali dengan node Aqua, Aqua setelah node Null dilanjutkan node Popok dan Sampo.
 6. Lintasan selanjutnya kembali pada node Gula dengan *support count* 4, Kopi dengan *support count* 3 dan Sabun dengan *support count* 2.
 7. Node Sprei, Celana dan Boneka berada pada node awal setelah node Null.
 8. Lintasan kedelapan diawali dengan node Aqua, node Aqua sudah ada sebelumnya sehingga hanya menambahkan *support count* menjadi 2 dan berikutnya adalah node Popok dengan *support count* 2, kemudian selanjutnya node Sprei, Sabun, Sampo dan Kemeja setelah node Popok.
 9. Node Gula bertambah *support count* sehingga menjadi 5 dan node Kopi menjadi 4 serta dalam node Kopi bertambah node Celana.
 10. Lintasan kesepuluh node Aqua bertambah 1 untuk *support count* sehingga menjadi 3 dan mempunyai node turunan dengan nama Sampo.
 11. Berikutnya node Gula dan Kopi bertambah lagi untuk *support count*, kemudian node Kopi mempunyai

node turunan Aqua dengan *support count* 1.

12. Lintasan yang terakhir adalah node Sprei dengan *support count* menjadi 2, berikutnya node Sampo dan Sabun.

Penerapan FP-Growth

Setelah tahap pembentukan FP-Tree, selanjutnya adalah penerapan algoritma FP-Growth untuk mencari *frequent itemset* yang signifikan. Seperti pada penjelasan sebelumnya, penerapan algoritma FP-Growth mengurutkan tiga langkah seperti : *conditional pattern base*, *conditional FP-Tree* dan *frequent pattern generated*. Berikut tabel yang menunjukkan masing-masing dari langkah penerapan algoritma FP-Growth:

Tabel 4
Conditional Pattern Base

Conditional Pattern Base
Kemeja : {{GL, KP, SR, PP, CL, BN :1}, {AQ, SR, SP, PP, SB :1}}
Boneka : {{GL, KP, SR, PP, CL :1}, {SR, CL:1}, {SR,PP}}
Celana : {{GL, KP, SR, PP:1}, {GL, KP :1}, {SR:1}}
Sabun : {{GL, KP, SP :1}, {GL, KP :1}, {AQ, SR, SP, PP :1}, {SR, SP :1}}
Popok : {{GL, KP, SR :1}, {SR :1}, {AQ, SP :1}, {AQ, SR, SP :1}}
Sampo : {{GL, KP :1}, {AQ :2}, {AQ, SR :1}, {SR :1}}
Sprei : {{GL, KP :1}, {AQ :1}}
Aqua : {{GL :1}, {GL, KP :1}}
Kopi : {GL : 5}

Tabel 5
Conditional FP-Tree

Conditional FP Tree
Kemeja : (GL :1, KP : 1), (AQ : 1, SR :1)
Boneka : (GL :1, KP : 1), (SR :2)
Celana : (GL : 2, KP :2), (SR :1)
Sabun : (GL :2, KP :2), (SR : 1), (AQ :1)
Popok : (SR :1, AQ :2)
Sampo : (GL :1, KP :1),
Sprei : (GL : 1, KP :2), (SR :1)
Aqua : (GL :2, KP :2)
Kopi : (GL :1)

Tabel 6
Frequent Pattern Generated

Pattern Generated
Kemeja : {GL, KM : 1}, {KP, KM : 1}, {AQ, KM : 1}
Boneka : {GL, BN : 1}, {SR, BN : 1}
Celana : {GL, CL : 2}, {KP, CL : 2}
Sabun : {GL, SB : 2}, {KP, SB : 2}, {SR, SB : 1}, {AQ, SB : 1}
Popok : {GL, PP : 1}, {KP, PP : 1}, {SR, PP : 1}, {AQ, PP : 1}
Sampo : {GL, SP : 1}, {KP, SP : 1}, {AQ, SP : 2}, {SR, SP : 1}
Sprei : {GL, SR : 1}, {KP, SR : 1}, {AQ, SR : 1}
Aqua : {GL, AQ : 2}, {KP, AQ : 2}
Kopi : {GL, KP : 5}

Penerapan Association Rule

Setelah pembahasan FP-Tree dan FP-Growth selanjutnya adalah penerapan *association rule* pada market basket.

Untuk mencari *association rule* adalah dengan mengelompokkan

sejumlah item yang disebut dengan itemset. Itemset adalah kumpulan item di keranjang belanja dalam jumlah kosong atau lebih (Prasetyo: 2012). Pada tabel dibawah ini adalah gambaran dari 1-itemset

Tabel 7
Itemset dengan 1-itemset

Frequent 2 Item	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	6	5	2	1	1	2	1	1	2	1
Kopi	5	5	1	1	1	2	1	1	2	1
Aqua	2	1	5	2	1	1	3	1	0	0
Popok	1	1	2	4	3	1	2	2	1	2
Sprei	1	1	1	3	5	2	2	2	2	3
Sabun	2	2	1	1	2	4	3	1	0	0
Sampo	1	1	3	2	2	3	5	1	0	0
Kemeja	1	1	1	2	2	1	1	2	1	1
Celana	2	2	0	1	2	0	0	1	3	2
Boneka	1	1	0	2	3	0	0	1	2	3

Untuk menghitung 2-itemset adalah dengan mencari produk yang terdiri dari 2 produk pada data transaksi, contoh yang pertama adalah produk {Gula, Gula} berjumlah 6 karena pembelian produk Gula ada 6 pada data transaksi, kemudian produk {Gula, Kopi}

berjumlah 5 karena pada data transaksi ada 5 pembelian produk Gula dan Kopi, selanjutnya produk {Gula, Aqua} berjumlah 2 karena ada 2 transaksi pembelian produk tersebut dan seterusnya sampai semua data transaksi dihitung.

Tabel 8
Itemset dengan 2-itemset.

Customer	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
1	1	1	0	0	0	1	1	0	0	0
2	1	1	0	1	1	0	0	1	1	1
3	0	0	0	1	1	0	0	0	0	1
4	1	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	1	0	0	0
6	1	1	0	0	0	1	0	0	0	0
7	0	0	0	0	1	0	0	0	1	1
8	0	0	1	1	1	1	1	1	0	0
9	1	1	0	0	0	0	0	0	1	0
10	0	0	1	0	0	0	1	0	0	0
11	1	1	1	0	0	0	0	0	0	0
12	0	0	0	0	1	1	1	0	0	0

Data selanjutnya adalah menghitung *support* dari masing-masing item, pada tabel dibawah ini adalah perhitungan *support* untuk 2-itemset menggunakan rumus *support* (1) dan rumus *confidence* (2). Contoh produk {Gula, Kopi}, untuk mencari *support* adalah dengan menghitung transaksi yang pembeliannya berisi produk {Gula, Kopi}. Data tersebut ada sebanyak 5,

kemudian dibagi jumlah transaksi 12 sehingga 5 dibagi 12 adalah 0.4167. Sedangkan menghitung *confidence* adalah dengan menghitung jumlah pembelian produk {Gula, Kopi} sebanyak 5 dibagi dengan jumlah pembelian Gula sebanyak 6 sehingga 5 dibagi 6 adalah 0.833 dan seterusnya sampai semua hubungan ada nilai *support* dan *confidence*.

Tabel 9
Support 2-itemset

Support (2 Items)	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	0,5	0,417	0,17	0,083	0,083	0,167	0,0833	0,0833	0,1667	0,08333
Kopi	0,42	0,417	0,08	0,083	0,083	0,167	0,0833	0,0833	0,1667	0,08333
Aqua	0,17	0,083	0,42	0,167	0,083	0,083	0,25	0,0833	0	0
Popok	0,08	0,083	0,17	0,333	0,25	0,083	0,1667	0,1667	0,0833	0,16667
Sprei	0,08	0,083	0,08	0,25	0,417	0,167	0,1667	0,1667	0,1667	0,25
Sabun	0,17	0,167	0,08	0,083	0,167	0,333	0,25	0,0833	0	0
Sampo	0,08	0,083	0,25	0,167	0,167	0,25	0,4167	0,0833	0	0
Kemeja	0,08	0,083	0,08	0,167	0,167	0,083	0,0833	0,1667	0,0833	0,08333
Celana	0,17	0,167	0	0,083	0,167	0	0	0,0833	0,25	0,16667
Boneka	0,08	0,083	0	0,167	0,25	0	0	0,0833	0,1667	0,25

Berikutnya adalah *confidence* untuk 2-itemset.

Tabel 10
Confidence 2-items

Confidence	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
Gula	1	0,833	0,33	0,167	0,167	0,333	0,1667	0,1667	0,3333	0,16667
Kopi	1	1	0,2	0,2	0,2	0,4	0,2	0,2	0,4	0,2
Aqua	0,4	0,2	1	0,4	0,2	0,2	0,6	0,2	0	0
Popok	0,25	0,25	0,5	1	0,75	0,25	0,5	0,5	0,25	0,5
Sprei	0,2	0,2	0,2	0,6	1	0,4	0,4	0,4	0,4	0,6
Sabun	0,5	0,5	0,25	0,25	0,5	1	0,75	0,25	0	0
Sampo	0,2	0,2	0,6	0,4	0,4	0,6	1	0,2	0	0
Kemeja	0,5	0,5	0,5	1	1	0,5	0,5	1	0,5	0,5
Celana	0,67	0,667	0	0,333	0,667	0	0	0,3333	1	0,66667
Boneka	0,33	0,333	0	0,667	1	0	0	0,3333	0,6667	1

PENUTUP

Kesimpulan

Penerapan algoritma FP-Growth dalam penelitian ini dapat diambil kesimpulan sebagai berikut :

1. Algoritma FP-Growth lebih efisien dalam memproses market basket atau data keranjang belanja konsumen.
2. Menggunakan *association rule* dapat mengetahui rule-rulebelanja konsumen seperti hubungan-hubungan tiap produk yang dibeli oleh konsumen dan mempengaruhi tipe-tipe konsumen dengan produk yang dibeli
3. *Association Rule* dapat memberikan keputusan pada manajemen untuk display produk, promosi, strategi pemasaran dan lain-lain

REFERENSI

- Aprilla, D. C. (2013). *Data Mining dengan Rapid Miner*. Jakarta.
- Jiawei han, M. K. (2012). *Data Mining Concepts and Techniques*. Waltham: Elsevier.

Prasetyo, E. (2012). *Data Mining Konsep dan Aplikasi Menggunakan MATLAB*. Yogyakarta: Penerbit ANDI Yogyakarta.

Samuel, D. (2008). Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset. 1-6.

Witten, I. H. (2011). *Data Mining Practical Machine Learning Tools and Techniques*. Burlington: Elsevier.

