

Analisis *Market Basket* Dengan Algoritma *Apriori* dan *FP-Growth*

Erwin*, Jurusan Teknik Informatika, *Fakultas Ilmu Komputer, Universitas Sriwijaya*

Abstrak—Algoritma yang umum digunakan dalam proses pencarian *frequent itemsets* (data yang paling sering muncul) adalah *Apriori*. Tetapi algoritma *Apriori* memiliki kekurangan yaitu membutuhkan waktu yang lama dalam proses pencarian *frequent itemsets*. Untuk mengatasi hal tersebut maka digunakanlah algoritma *FP-Growth*. Dalam makalah ini akan dibahas penerapan *Apriori* dan *FP-Growth* dalam proses pencarian *frequent itemsets*. Penggunaan *FP-Tree* yang digunakan bersamaan dengan algoritma *FP-growth* untuk menentukan *frequent itemset* dari sebuah database, berbeda dengan paradigma *Apriori* yang memerlukan langkah *candidate generation*, yaitu dengan melakukan *scanning database secara berulang-ulang* untuk menentukan *frequent itemset*. Makalah ini juga menyajikan pembahasan mengenai perbandingan kompleksitas waktu antara algoritma *FP-growth* dengan *Apriori* dan hasil dari perbandingan algoritma tersebut.

Kata Kunci—Data Mining, Algoritma *Apriori*, *FP-Tree*, *Frequent Pattern Mining*, *Frequent Itemset*

I. PENDAHULUAN

Inventori (stok barang) merupakan permasalahan operasional yang sering dihadapi oleh swalayan. Jika jumlah inventori terlalu sedikit dan permintaan tidak dapat dipenuhi karena kekurangan persediaan, maka akan mengakibatkan konsumen merasa kecewa dan ada kemungkinan konsumen tidak akan kembali lagi. Begitu juga jika inventori terlalu besar, maka akan mengakibatkan kerugian bagi swalayan karena harus menyediakan tempat yang lebih besar, terjadinya penyusutan nilai guna barang, serta harus menyediakan biaya tambahan yang terkait dengan biaya inventori seperti biaya pemeliharaan dan biaya akuntansi.

Salah satu cara yang bisa dilakukan untuk mengetahui kondisi pasar (konsumen) adalah dengan mengamati data transaksi penjualan. Data transaksi penjualan disimpan dalam basis data server dalam jumlah yang sangat besar. Data inilah yang kemudian diolah sehingga dihasilkan laporan penjualan dan laporan laba rugi swalayan. Akan tetapi, data penjualan tersebut bisa diolah lebih lanjut sehingga didapatkan informasi baru. Teknologi *data mining* hadir sebagai solusi nyata bagi para pengambil keputusan seperti manajer dalam menentukan strategi pemasaran dan keterkaitan antara barang yang dibeli oleh konsumen sehingga dapat meningkatkan pelayanan pada konsumen.

Tujuan dari *data mining* yaitu untuk menemukan

hubungan atau pola-pola yang memberikan manfaat ke pada pihak perusahaan atau organisasi. *Data mining* menggunakan pendekatan *discovery based* yaitu dimana pencocokan pola (*pattern-matching*) dan algoritma digunakan untuk menentukan relasi-relasi kunci di dalam data yang akan dieksplorasi.

Pola merupakan instansiasi dari model. *Data mining* melakukan pencocokan model atau menentukan pola dari data yang diobservasi. Sebuah pola membagi sekumpulan data dan melibatkan aspek spasial yang mungkin divisualisasi. Pola tersebut harus bersifat sah (*valid*), baru, dapat bermanfaat (*potentially usefull*) dan dapat dimengerti (*ultimately understandable*).

II. TINJAUAN PUSTAKA

A. Algoritma *Apriori*

Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian *frequent itemset* dengan menggunakan teknik *association rule* [10]. Algoritma *Apriori* menggunakan knowledge mengenai *frequent itemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Pada algoritma *Apriori* untuk menentukan kandidat-kandidat yang mungkin muncul dengan cara memperhatikan minimum support.

Adapun dua proses utama yang dilakukan dalam algoritma *Apriori*, yaitu [3] :

1. *Join* (penggabungan).

Pada proses ini setiap item dikombinasikan dengan item yang lainnya sampai tidak terbentuk kombinasi lagi.

2. *Prune* (pemangkasan).

Pada proses ini, hasil dari item yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan minimum support yang telah ditentukan oleh user.

Dua proses utama tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*, yang dapat dilihat pada Gambar 1 berikut ini.

Walaupun algoritma *Apriori* mudah untuk dipahami dan diimplementasikan dibandingkan dengan algoritma yang lainnya yang memang diterapkan untuk proses *association rule*, akan tetapi algoritma *Apriori* juga memiliki kekurangan yaitu, untuk melakukan pencarian *frequent itemset*, algoritma *Apriori* harus melakukan *scanning database* berulang kali untuk setiap kombinasi item. Hal tersebut menyebabkan banyaknya waktu yang dibutuhkan untuk melakukan *scanning database*. Selain itu, dibutuhkan *generate*

*Erwin adalah peneliti dan staf pengajar di Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya, Ogan Ilir, Sumatera Selatan. (E-mail: erwin@unsri.ac.id).

candidate yang besar untuk mendapatkan kombinasi *item* dari *database*.

```

Input :
D, a database of a transactions;
Min_support, the minimum support count
threshold
Output : L, frequent itemsets in D
Mtehod :
L1 = find_frequent_1_itemsets(D);
for (k = 2; Lk-1 ≠ ∅; k | |) {
    Ck = Apriori_gen(Lk-1);
    for each transaction t ⊂ D { //scan D for
counts
        Ct = subset (Ck,t); //get the subsets
of t that are candidates
        for each candidate c ∈ Ct
            c.count | |;
        }
    Lk = {c ⊂ Ck | c.counts ≥ min_sup}
}
return L = ∪k Lk;

```

Gambar 1. Algoritma *Apriori*

B. Algoritma *FP-Growth*

Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*. Sehingga kekurangan dari algoritma *Apriori* diperbaiki oleh algoritma *FP-Growth*.

Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data [14].

Pada algoritma *Apriori* diperlukan *generate candidate* untuk mendapatkan *frequent itemsets*. Akan tetapi, di algoritma *FP-Growth generate candidate* tidak dilakukan karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemsets*. Hal tersebutlah yang menyebabkan algoritma *FP-Growth* lebih cepat dari algoritma *Apriori*.

Karakteristik algoritma *FP-Growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-Tree*. Dengan menggunakan *FP-Tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent Itemset* dari *FP-Tree*.

Penggalian itemset yang *frequent* dengan menggunakan algoritma *FP-Growth* akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan *FP-Tree*. Metode *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai [3]:

1. Tahap pembangkitan *conditional pattern base*,
2. Tahap pembangkitan *conditional FP-Tree*, dan
3. Tahap pencarian *frequent itemset*.

Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*, yang dapat dilihat pada algoritma berikut :

```

Input : FP-Tree Tree
Output : Rt sekumpulan lengkap pola
frequent
Method : FP-growth (Tree, null)
Procedure : FP-growth (Tree, α)
{

```

```

01: if Tree mengandung single path P;
02: then untuk tiap kombinasi (dinotasikan β)
dari node-node dalam path do
03: bangkitkan pola β α dengan support dari
node-node dalam β;
04: else untuk tiap a1 dalam header dari Tree
do
{
05: bangkitkan pola
06: bangun β = a1 α dengan support = a1.
support
07: if Tree β = ∅
08: then panggil FP-growth (Tree, β)
}
}

```

Gambar 2. Algoritma *FP-Growth*

III. EKSPERIMEN

Data utama yang digunakan pada penelitian ini adalah data transaksi yang terdiri dari 5.000 transaksi pada tahun 2009. Karena pada penelitian ini adalah menganalisis perbandingan antara *Apriori* dan *fp-growth* maka data yang digunakan tersebut akan dibuat menjadi dua tetapi dengan jumlah data transaksi yang berbeda yang dapat dilihat pada Tabel 1.

TABEL 1. SPESIFIKASI DATA UNTUK UJI COBA

	Data 1	Data 2
Nama Tabel	Trans1	Trans2
Jml Record	8.478	16.731
Jml Transaksi	2.500	5.000
Jml Item	1.794	1.794

Sebelum proses *data mining* dimulai, dilakukan proses *preprocessing* sehingga didapatkan *dataset* seperti pada Tabel 2.

TABEL 2. DATASET YANG DIHASILKAN SETELAH MELALUI *PREPROCESSING*

No Bon	Item
1	Biskuat Susu Krim Vanila 57gr
1	Ciptadent Flexi Grip Medium
1	Pantene Shampoo Smooth&Silky 90 MI
2	Oops Butter Crackers Real Butter 26
2	Vape Reff 720 MI
3	Bendera Bubuk Instant Coklat 200 Gr
3	Sariwangi Tea Bag Asli 25 Gr
3	Pocari Sweat Can 330 MI
3	Fanta Grape 330 MI
.....

Setelah melalui proses *preprocessing*, dataset tersebut bisa di *mining* dengan menggunakan algoritma asosiasi yaitu *Apriori* dan *FP-growth*.

Untuk algoritma *Apriori*, prosesnya yaitu dataset di-*scan*, kemudian setiap item dikombinasikan dengan item yang lainnya lalu dengan menggunakan minimum *support* kombinasi item tersebut dipangkas untuk menghasilkan itemsets yang nantinya akan dikombinasikan lagi sampai

tidak dapat terbentuk lagi kombinasi item berdasarkan nilai minimum support yang telah ditentukan oleh user. Untuk dataset dengan jumlah transaksi 2.500 dapat dilihat pada Gambar 3, sedangkan untuk dataset dengan jumlah transaksi 5.000 dapat dilihat pada Gambar 4.

No Bon	Item
1	Biskuat Susu Krim Vanila 57gr
1	Ciptadent Flexi Grip Medium
....
43	DAIA Lemon 380 Gr
43	Giv Merah Soap Kotak 80 Gr
44	Vape Reff 720 Ml
.....

Itemsets	Support
Mizone lemon	20
Tolak Angin Cair	20
Happy Juice Orange	20
Sgm 1 600 Gr	21
..	..

Itemsets	Supp
Mizone lemon, Tolak Angin Cair	19
Mizone lemon, Happy Juice Orange	20
Mizone lemon, Sgm 1 600 Gr	23
Happy Juice Orange, Sgm 1 600 Gr	23
..	..

Itemsets	Supp
Mizone lemon, Happy Juice Orange	20
Mizone lemon, Sgm 1 600 Gr	23
Happy Juice Orange, Sgm 1 600 Gr	23
..	..

Gambar 3. Pembangkitan Kaidah *Itemsets* dan *Large Itemsets* dengan Jumlah Transaksi 2.762

No Bon	Item
1	Biskuat Susu Krim Vanila 57gr
1	Ciptadent Flexi Grip Medium
....
43	DAIA Lemon 380 Gr
43	Giv Merah Soap Kotak 80 Gr
44	Vape Reff 720 Ml
.....

Itemsets	Supp
Fugu seaweed 22gr	40
Sprite 1500 ml	40
Frutang hot fill 330 ml	20
Boom jeruk nipis	41
..	..

Itemsets	Supp
Fugu seaweed 22gr, Sprite 1500 ml	40
Fugu seaweed 22gr, Frutang hot fill 330 ml	40
Fugu seaweed 22gr, Boom jeruk nipis	40
Frutang hot fill 330 ml, Boom jeruk nipis	41
..	..

Itemsets	Supp
Mizone lemon, Sgm 1 600 Gr	43
Happy Juice Orange, Sgm 1 600 Gr	43
..	..

Gambar 4. Pembangkitan Kaidah *Itemsets* dan *Large Itemsets* dengan Jumlah Transaksi 5.000

Proses pertama kali untuk mencari *frequent itemsets* sama dengan algoritma *Apriori*. Kumpulan *frequent itemsets* dicari dengan melakukan *scanning database* kemudian hasilnya diakumulasikan dengan tiap *items* lainnya dan dikelompokkan untuk mendapatkan *frequent itemsets* yang memenuhi nilai minimum support.

Setelah didapatkan *frequent itemsets*, lalu *frequent itemsets* yang memiliki nilai kurang dari minimum support akan dipangkas atau dibuang. Sehingga akan didapatkan *frequent itemsets* yang memenuhi minimum support dan disusun dengan urutan menurun. Hasil dari list tersebut dinotasikan dengan L.

Setelah itu, dilakukan *scanning database* untuk kedua kalinya. Pada proses ini *FP-Tree* mulai dibentuk. Pertama, membuat akar dari pohon dan diberi nama null. Lalu, L dipindai untuk membuat batang dari setiap transaksi yang ada di L. Pembacaan transaksi pertama yang ada di L akan membuat simpul, sehingga akan terbentuk lintasan transaksi. *Support count* dari setiap transaksi bernilai satu. Kemudian, dilanjutkan proses pembacaan transaksi kedua.

Dari pembangunan *FP-Tree*, maka dapat diterapkan algoritma *FP-Growth*. Untuk menemukan *frequent itemset*, maka perlu ditentukan upapohon dengan lintasan yang berakhir dengan *support count* terkecil. Nantinya akan didapatkan *frequent itemsets* berdasarkan *suffix* yang dapat dilihat pada Tabel 3 dan Tabel 4, yaitu :

TABEL 3. HASIL *FREQUENT ITEMSETS* DENGAN JUMLAH TRANSAKSI 2.500

<i>Suffix</i>	<i>Frequent Itemsets</i>
DAIA Lemon 380 Gr	{ Vape Reff 720 ml, DAIA Lemon 380 Gr }, { Giv Merah Soap Kotak 80 Gr,

	DAIA Lemon 380 Gr }, {Vape Reff 720 ml, Giv Merah Soap Kotak 80 Gr, DAIA Lemon 380 Gr }
Giv Merah Soap Kotak 80 Gr	{Vape Reff 720 ml, Giv Merah Soap Kotak 80 Gr}, { }
..	..

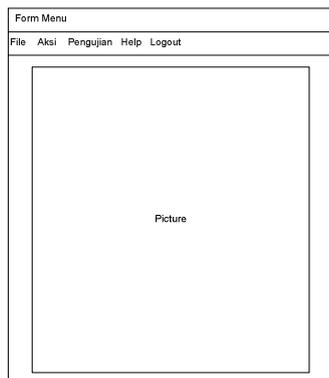
TABEL 4. HASIL *FREQUENT ITEMSETS* DENGAN JUMLAH TRANSAKSI 5.000

<i>Suffix</i>	<i>Frequent Itemsets</i>
...

IV. RANCANGAN ANTARMUKA APLIKASI

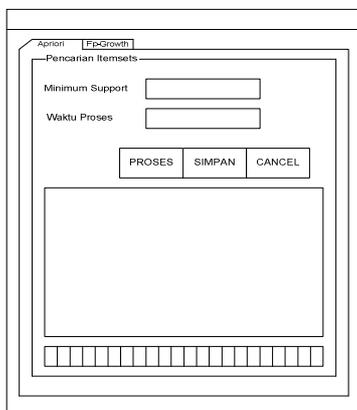
Dalam penelitian ini, kami menggunakan aplikasi Borland Builder C++. Berikut adalah rancangan antar muka yang akan dibuat

1) Rancangan Antar Muka Menu Utama



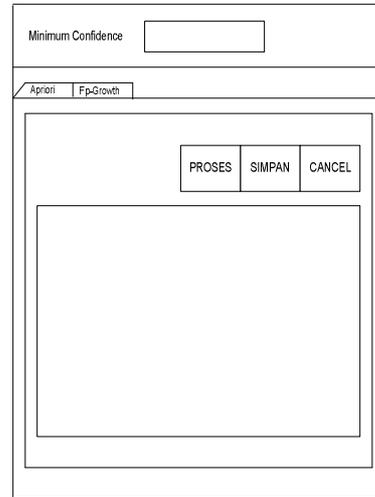
Gambar 5. Antarmuka Menu Utama

2) Rancangan Antar Muka Pencarian *Itemsets*



Gambar 6. Antar Muka Pencarian *Itemsets*

3) Rancangan Antar Muka Menu Pembentukan *Rule*



Gambar 7. Antar Muka Menu Pembentukan *Rule*

V. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah dengan menjalankan perangkat lunak dengan parameter yang sama pada tabel yang berbeda. Tabel yang akan digunakan ada dua, yaitu: Trans1, dan Trans2. Sedangkan parameter yang digunakan adalah minsup dan minconf. Hasil uji coba untuk masing-masing tabel dapat dilihat pada Tabel 5 dan Tabel 6.

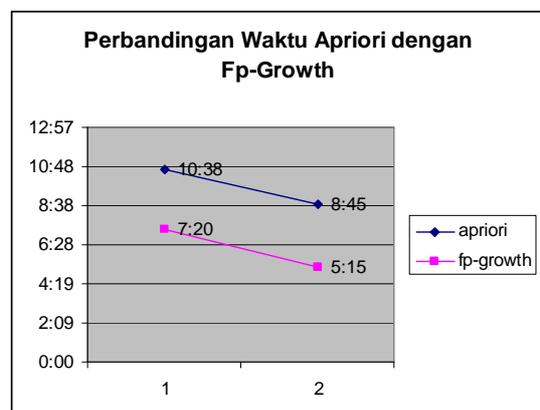
TABEL 5. HASIL UJI COBA 1 MENGGUNAKAN DATA 1

	<i>Apriori</i>	<i>FP-Growth</i>
Min Sup	20	20
Waktu proses	10 jam 38 menit	7 Jam 20 menit

TABEL 6. HASIL UJI COBA 1 MENGGUNAKAN DATA 2

	<i>Apriori</i>	<i>FP-Growth</i>
Min Sup	40	40
Waktu proses	8 jam 45 menit	5 jam 15 menit

Dari kedua hasil coba di atas, dibuat suatu grafik berdasarkan nilai support dan waktu yang dibutuhkan untuk mencari *frequent k-itemset (Process Time)*. Grafik tersebut dapat dilihat pada Gambar 8.



Gambar 8. Grafik Perbandingan Waktu *Apriori* dengan *FP-Growth*

VI. KESIMPULAN

Beberapa kesimpulan yang dapat ditarik dari penulisan makalah ini adalah algoritma *Apriori* membutuhkan waktu komputansi yang lama untuk mendapatkan *frequent itemsets*. Karena berulang kali melakukan pemindaian data. Selain itu algoritma ini membutuhkan alokasi memori yang besar untuk melakukan pencarian *itemsets*.

FP-Tree yang terbentuk dapat memampatkan data transaksi yang memiliki item yang sama, sehingga penggunaan memori komputer lebih sedikit, dan proses pencarian *frequent itemset* menjadi lebih cepat. *FP-Growth* hanya membutuhkan dua kali *scanning database* dalam mencari *frequent itemsets* sehingga waktu yang dibutuhkan pun menjadi relatif singkat dan efisien.

REFERENSI

- [1] Arini, et al. 2007. Analisis Kinerja Algoritma FOLD Growth dan FP-Growth Pada Pengalian Pola Asosiasi. [Online] Tersedia : www.si.its.ac.id/penelitian/JURNAL/Arin.pdf [10 Maret 2009]
- [2] Gregorius S. Budhi ST., MT. et al. 2003. Aplikasi Data Mining Dengan Konsep Fuzzy c-Covering Untuk Analisa Market Basket Pada Pasar Swalayan.
- [3] Han Jiawei, and M. Kamber. 2006. Data Mining: Concepts and Techniques, Morgan Kaufmann, USA.
- [4] Kokoh, Philips. 2006. Association Rule Mining. [Online] Tersedia : <http://philips.wordpress.com/2006/05/10/association-rule-mining/>. [1 April 2009]
- [5] Krutchen, P. 2000. The Rational Unified Process An Introduction, Second Edition. Addison Wesley Professional, USA.
- [6] Kusnawi, 2007. Pengantar Solusi data mining. STMIK AMIKOM Surabaya.
- [7] Larman, C. 2004. Applying UML and Patterns : An Introduction to Object Oriented Analysis and Design and Iterative Development, Third Edition. Addison Wesley Professional, USA.
- [8] M.J. Berry and G. Linoff. 1997. Data Mining Techniques for Marketing, Sales and Customer Support, John Wiley and Sons.
- [9] Moertini, V.S. 2002. Data Mining Sebagai Solusi Bisnis. [Online] Tersedia: http://home.unpar.ac.id/~integral/Volume%207_idatamining_ok.pdf. [18 Desember 2008]
- [10] Moertini, Veronika dan Marsela Yulita. 2007. Analisis Keranjang Pasar Dengan Algoritma Hash-Based Pada Data Transaksi Penjualan Apotek. Jurusan Ilmu Komputer, Universitas Katolik Parahyangan, Bandung.
- [11] Prayitno, Usman, et al. 2004. Aplikasi Data Mining dengan Konsep Fuzzy c-Covering untuk Analisa Market Basket pada pasar swalayan. [Online] Tersedia : www.petra.ac.id/~puslit/journals/request.php. [1 April 2009]
- [12] Palace, Bill. 1996. Data Mining: What Is Data Mining?. [Online] Tersedia: www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm. [1 April 2009]
- [13] Pei, J., Han, J., Mortazavi, B., Pinto, H. 2001. PrefixSpan : Mining Sequential Patterns Efficiently By Prefix-Projected Pattern Growth. Intelligent database Systems Research Lab Report. School of Computing Science, Simon Fraser University, Burnaby, B. C., Canada.
- [14] Samuel, David. 2008. Penerapan Struktur *FP-Tree* dan Algoritma FP-Growth dalam Optimalisasi Penentuan Frequent Itemset. Institut Teknologi Bandung.
- [15] Willyanto, Leo. 2005. Pembuatan Perangkat Lunak Data Mining Untuk Penggalan kaidah Asosiasi Menggunakan Metode Apriori. [Online] tersedia : www.petra.ac.id/~puslit/journals/request.php. [1 April 2009]