

PENERAPAN *IMPROVED APRIORI* PADA APLIKASI *DATA MINING* DI PERUSAHAAN KALVIN SOCKS PRODUCTION

Yepi Septiana¹, Dian Dharmayanti²

Teknik Informatika - Universitas Komputer Indonesia

Jl. Dipatiukur 112-114 Bandung

Email : yepi.septiana@gmail.com¹, diandmdr@yahoo.com²

ABSTRAK

Algoritma *apriori* merupakan algoritma klasik yang sering digunakan. Kekurangan yang ada pada algoritma *apriori* adalah harus melakukan *scanning* berulang terhadap keseluruhan *database* tiap kali *iterasi*. Semakin banyak data transaksi yang akan diproses maka semakin lama juga waktu yang dibutuhkan.

Kalvin Socks Production merupakan salah satu perusahaan yang menggunakan teknologi *data mining* dengan algoritma *apriori* untuk mencari pola pembelian dari para pelanggannya. Banyaknya data transaksi penjualan mengakibatkan proses dalam pencarian pola pembelian membutuhkan waktu yang cukup lama.

Improved apriori mempresentasikan *database* ke dalam bentuk *matrix* untuk menggambarkan relasi dalam *database*. Kemudian *matrix* dihitung untuk mencari nilai *support* dari *candidate frequent itemset* yang memenuhi kriteria untuk menghasilkan *frequent itemset* tanpa melakukan *scanning* ulang terhadap *database*. Salah satu cara untuk mengatasi masalah yang ada pada algoritma *apriori* adalah dengan menggunakan algoritma *improved apriori*.

Kata kunci : *Data mining, association rule, algoritma improved apriori, frequent itemset.*

1. Pendahuluan

1.1. Latar Belakang

Banyak penelitian dilakukan untuk mencari informasi, inovasi baru ataupun lainnya. Salah satunya penelitian di Calvin Socks Production. Penelitian tersebut dilakukan untuk mencari dan menggali informasi lebih dari para pelanggannya melalui data transaksi penjualan. Dengan adanya data transaksi penjualan Calvin Socks Production menggunakan teknologi *data mining* untuk mencari pola pembelian dari pelanggannya dan mencari *item* mana saja yang sering dipesan secara bersamaan oleh pelanggannya. Penelitian di Calvin Socks Production dilakukan oleh Deasy Rusmawati dengan judul “Penerapan Data Mining Pada Penjualan Kaos Kaki Di Pabrik Calvin Socks Production Menggunakan Metode Association” dan hasil dari

penelitian tersebut menghasilkan suatu pola pembelian dari pelanggan Calvin Socks Production, lalu dari pola tersebut didapatlah sebuah informasi mengenai jenis-jenis kaos kaki mana saja yang sering dipesan secara bersamaan oleh pelanggan Calvin Socks Production, yang mana dari informasi tersebut digunakan oleh pihak perusahaan dalam mempertimbangkan jenis kaos kaki yang akan diproduksi lebih bulan selanjutnya [1].

Aplikasi hasil dari penelitian tersebut sudah digunakan oleh pihak perusahaan sejak 4 bulan yang lalu. Namun aplikasi tersebut memiliki sebuah kekurangan. Hasil uji coba aplikasi pada data transaksi penjualan bulan Mei 2015 dengan total data transaksi sebanyak 176 data transaksi, 4 nilai *minimum support*, dan 40% nilai *minimum confidence* membutuhkan waktu sekitar 25 menit. Padahal pihak perusahaan sudah menggunakan spesifikasi komputer sesuai dengan standar kebutuhan. Itu terjadi karena algoritma *apriori* harus melakukan *scan database* setiap kali *iterasi*-nya. Selain itu lamanya waktu yang di butuhkan untuk pemrosesan tersebut tergantung dari nilai *minimum support* dan *minimum confidence* yang diinputkan. Semakin kecil nilai *minimum support* dan *minimum confidence* maka semakin lama pula waktu yang dibutuhkan karena akan terdapat banyak kombinasi *item*-nya.

Association rule adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*. Terdapat banyak algoritma yang ada pada *Association rule* seperti algoritma *apriori, fg growth, ct-pro, improved apriori* dll. Salah satu cara untuk mengatasi masalah yang terjadi pada *apriori* adalah dengan menggunakan *improved apriori algorithm*. *Improved apriori* mempresentasikan *database* ke dalam bentuk *matrix* untuk menggambarkan relasi dalam *database*. Kemudian *matrix* dihitung untuk mencari nilai *support* dari *candidate frequent itemset* yang memenuhi kriteria untuk menghasilkan *frequent itemset* tanpa melakukan *scanning* ulang terhadap *database* dengan menggunakan operasi “AND” terhadap baris *matrix* sesuai dengan *item* dalam *candidate frequent itemset* [2].

2. Tinjauan Pustaka

2.1. Data Mining

“Data mining adalah proses menemukan pengetahuan yang menarik dari sejumlah data yang besar yang disimpan di dalam database, gudang data, atau repositori informasi lainnya” [6].

kemajuan luar biasa yang terus berlanjut dalam bidang data mining didorong oleh beberapa faktor, antara lain :

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data warehouse, sehingga seluruh perusahaan memiliki akses ke dalam database yang andal.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetensi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk data mining (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Dari definisi-definisi yang telah disampaikan, hal penting yang terkait dengan data mining adalah :

1. Data mining merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan data mining adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

2.2. Cross- Industry Standard Process for Data Mining (CRISP- DM)

Cross- Industry Standard Process for Data Mining (CRISP- DM) yang dikembangkan tahun 1996 oleh analis dari beberapa industry seperti Daimler Chrysler, SPSS, dan NCR. CRISP DM menyediakan standar proses data mining sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP- DM, sebuah proyek data mining memiliki siklus hidup yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antar fase digambarkan dengan panah. Sebagai contoh, jika proses berada pada fase modelling. Berdasar pada perilaku dan karakteristik model, proses mungkin harus kembali kepada fase data preparation untuku perbaikan lebih lanjut terhadap data atau berpindah maju kepada fase evaluation [5].

2.3. Asosiasi (Association)

Analisis asosiasi atau Association rule mining adalah teknik data mining untuk menemukan aturan asosiatif antara suatu kombinasi item.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu support dan confidence. Support (nilai penunjang) adalah presentase kombinasi item tersebut dalam database, sedangkan confidence (nilai kepastian) adalah kuatnya hubungan antar-item dalam aturan asosiasi.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap [6] :

1. Analisis pola frekuensi tinggi
Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai support dalam database. Nilai support sebuah item diperoleh dengan rumus berikut.

$$Support(A) = \frac{\text{Jumlah transaksi mengandung A}}{\text{Total Transaksi}} \quad (1)$$

Sementara itu, nilai support dari 2 item diperoleh dari rumus berikut.

$$Support(A, B) = P(A \cap B) \quad (2)$$

$$Support(A, B) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi}} \quad (3)$$

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang cukup kuat tingkat ketergantungan antar item dalam antecedent (pendahulu) dan consequent (pengikut) serta memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan Asosiatif $A \rightarrow B$.

Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D merepresentasikan himpunan item yang berada dalam I. I adalah himpunan item yang dijual. Misalkan kita memilih himpunan item A dan himpunan item lain B, kemudian aturan asosiasi akan berbentuk:

$$\text{jika A, maka B } (A \rightarrow B)$$

Dimana antecedent A dan consequent B merupakan subset dari I, dan A dan B merupakan mutually exclusive dimana aturan :

$$\text{jika A, maka B}$$

Tidak berarti

$$\text{jika B, maka A}$$

Sebuah itemset adalah himpunan item-item yang ada dalam I, dan k-itemset adalah itemset yang berisi k item. Frekuensi itemset merupakan itemset yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan (ϕ). Misalkan $\phi = 2$, maka semua itemset yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut frequent. Himpunan dari frequent k-itemset dilambangkan dengan F_k .

Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus berikut.

$$Confidence = P(B | A) = \frac{\Sigma \text{Transaksi mengandung A dan B}}{\Sigma \text{Transaksi mengandung A}} \quad (4)$$

2.4. Algoritma Improved Apriori

Algoritma *Improved Apriori* berbasis *Matrix* diusulkan oleh beberapa peneliti, namun dengan teknik yang berbeda saat pencarian *frequent itemset*. merepresentasikan *database* ke dalam bentuk *matrix* untuk menggambarkan relasi dalam *database*. Kemudian *matrix* dihitung untuk mencari nilai *support* dari *candidate frequent itemset* yang memenuhi kriteria untuk menghasilkan *frequent itemset* tanpa melakukan *scanning* ulang terhadap *database* dengan menggunakan operasi “AND” terhadap baris *matrix* sesuai dengan *item* dalam *candidate frequent itemset* dan menambahkan hasil dari AND, dengan hasilnya adalah *Support* [7]. Algoritma ini tidak melakukan *scan* ulang terhadap *database* untuk mencari hubungan seperti algoritma sebelumnya, maka waktu komputasi dan pencarian *candidate frequent itemset* menjadi lebih cepat.

Tahapan algoritma ini berjalan sebagai berikut [2]:

1. Konversi *database* ke dalam bentuk *matrix*.
 - a Konversi *database* yang berisi I_n item dan transaksi T_m ke dalam bentuk *matrix*. Baris dari *matrix* mewakili transaksi dan kolom dari *matrix* mewakili *item*. Jika pada suatu transaksi terdapat item maka nilainya adalah 1 dan bernilai 0 jika sebaliknya.
 - b Jumlah nilai dari kolom adalah nilai *support count* dan jumlah nilai dari baris adalah banyaknya item dalam suatu transaksi atau disebut *count*.
2. Periksa jumlah kolom dan jumlah baris.
 - a Hapus kolom yang jumlah kolomnya kurang dari nilai *minimum support*.
 - b Hapus baris yang jumlah barisnya kurang dari sama dengan nilai k (*k-frequent itemset*).
3. Gabungkan tiap kolomnya menggunakan *cross product* untuk menemukan kombinasi *frequent 2-itemset* dan gunakan operasi AND untuk mendapatkan nilainya.
4. Periksa jumlah kolom dan jumlah baris
 - a Hapus kolom yang jumlah kolomnya kurang dari nilai *minimum support*.
 - b Hapus baris yang jumlah barisnya kurang dari sama dengan nilai k (*k-frequent itemset*).
5. Demikian pula untuk mencari K_{th} -*frequent itemset*. Gabungkan tiap kolomnya dan hapus kolom yang kurang dari *minimum support* dan hapus baris yang jumlah barisnya kurang dari sama dengan k

3. Hasil Penelitian

3.1. Analisis Sumber Data

Data yang digunakan pada penelitian ini adalah data laporan transaksi penjualan kaos kaki periode Mei 2015 sampai Agustus 2015 .

Berikut sampel transaksi penjualan kaos kaki.

Tabel 1. Sampel data transaksi penjualan

tanggal	NoNota	kodeBarang	NamaBarang	jumlah
1/8/2015	4	B19	sd p kaoxin	6
1/8/2015	4	B20	smp p kaoxin	12
1/8/2015	4	B29	bayi lipat kids	12
5/8/2015	18	B17	smp hp kaoxin	36
5/8/2015	18	B18	sma hp kaoxin	12
5/8/2015	18	B19	sd p kaoxin	24
5/8/2015	18	B20	smp p kaoxin	12
5/8/2015	18	B22	mk sd h polos	12
5/8/2015	20	B15	sma h kaoxin	12
5/8/2015	20	B17	smp hp kaoxin	12
5/8/2015	20	B18	sma hp kaoxin	12
5/8/2015	20	B19	sd p kaoxin	12
5/8/2015	20	B20	smp p kaoxin	24
5/8/2015	20	B28	bayi lurus kids	12
20/8/2015	111	B15	sma h kaoxin	36
20/8/2015	111	B17	smp hp kaoxin	12
20/8/2015	111	B18	sma hp kaoxin	12
20/8/2015	111	B19	sd p kaoxin	12
20/8/2015	111	B20	smp p kaoxin	12
24/8/2015	142	B16	sd hp kaoxin	6
24/8/2015	142	B17	smp hp kaoxin	24
24/8/2015	142	B18	sma hp kaoxin	24

3.2. Analisis Preprocessing

Adapun langkah-langkah *preprocessing* data dalam penelitian ini adalah sebagai berikut :

1. Pemilihan Atribut (*atribut selection*)

Berdasarkan informasi yang ingin didapat oleh pengguna mengenai jenis produk yang dibeli secara bersamaan, maka dalam tahap pemilihan atribut ini, atribut yang akan digunakan dari data hasil ekstraksi adalah atribut NoNota dan kodeBarang. Atribut NoNota adalah ID dari transaksi dan atribut kodeBarang adalah kode dari jenis barang yang dibeli.

Tabel 2. Hasil pemilihan atribut

NoNota	kodeBarang
4	B19,B20,B29
18	B17,B18,B19,B20,B22
20	B15,B17,B19,B20,B28
111	B15,B18,B19,B20
142	B15,B16,B17,B18

2. Pembersihan Data (*data cleaning*)

Pada tahap pembersihan data, hasil pemilihan atribut akan dibersihkan dari data transaksi yang mengandung item tunggal. Data transaksi yang memiliki item tunggal ini tidak memiliki hubungan asosiasi dengan item lain yang sudah dibeli.

Tabel 3. Hasil Pembersihan Data

NoNota	kodeBarang
4	B19,B20,B29
18	B17,B18,B19,B20,B22
20	B15,B17,B19,B20,B28
111	B15,B18,B19,B20
142	B15,B16,B17,B18

3.3. Analisis Penerapan Metode Association Rule

Langkah-langkah proses pengerjaan algoritma *improved apriori* dalam penelitian ini adalah sebagai berikut :

1. Asumsi nilai *minimum support* yang akan digunakan adalah 3
2. Asumsi nilai *minimum confidence* yang akan digunakan sebesar 100%.
3. Dari hasil pembersihan data kemudian dilakukan transformasi ke dalam bentuk *matrix*. NoNota mewakili baris sedangkan kodeBarang mewakili kolom.

Tabel 4. Transformasi kedalam bentuk *matrix*

	B15	B16	B17	B18	B19	B20	B22	B28	B29
4	0	0	0	0	1	1	0	0	1
18	0	0	1	1	1	1	1	0	0
20	1	0	1	1	1	1	0	1	0
111	1	0	1	1	1	1	0	0	0
142	0	1	1	1	0	0	0	0	0

4. Kemudian untuk mencari *frequent 1-itemset* hapus jumlah kolom yang jumlah nilainya kurang dari *minimum support* dan hapus baris yang jumlah nilainya kurang dari sama dengan

Tabel 5. *Frequent 1-itemset*

	B17	B18	B19	B20
4	0	0	1	1
18	1	1	1	1
20	1	1	1	1
111	1	1	1	1
142	1	1	0	0

5. Untuk mendapatkan calon kandidat *2-itemset* maka lakukan *cross product* sedangkan untuk mendapatkan nilai *support*-nya gunakan operan AND.

Tabel 6. Kandidat *frequent 2-itemset*

	B17,B18	B17,B19	B17,B20	B18,B19	B18,B20	B19,B20
4	0	0	0	0	0	1
18	1	1	1	1	1	1
20	1	1	1	1	1	1
111	1	1	1	1	1	1
142	1	0	0	0	0	0

6. Data yang jumlah kolomnya kurang dari nilai *minimum support* akan dihapus dan baris yang jumlah nilainya kurang dari 2 akan di hapus

Tabel 7. *Frequent 2-itemset*

	B17,B18	B17,B19	B17,B20	B18,B19	B18,B20	B19,B20
18	1	1	1	1	1	1
20	1	1	1	1	1	1
111	1	1	1	1	1	1

7. Untuk mendapatkan calon kandidat *3-itemset* maka lakukan *cross product* sedangkan untuk mendapatkan nilai *support*-nya gunakan operan AND.

Tabel 8. kandidat *frequent 3-itemset*

	B17,B18,B19	B17,B18,B20	B17,B19,B20	B18,B19,B20
18	1	1	1	1
20	1	1	1	1
111	1	1	1	1

- Data yang jumlah kolomnya kurang dari nilai *minimum support* akan dihapus dan baris yang jumlah nilainya kurang dari 3 akan di hapus

Tabel 9. *Frequent 3-itemset*

	B17,B18,B19	B17,B18,B20	B17,B19,B20	B18,B19,B20
18	1	1	1	1
20	1	1	1	1
111	1	1	1	1

- Untuk mendapatkan calon kandidat *3-itemset* maka lakukan *cross product* sedangkan untuk mendapatkan nilai *support*-nya gunakan operan AND.

Tabel 10. kandidat *frequent 4-itemset*

	B17,B18,B19,B20
18	1
20	1
111	1

- Data yang jumlah kolomnya kurang dari nilai *minimum support* akan dihapus dan baris yang jumlah nilainya kurang dari 4 akan di hapus

Tabel 11. *Frequent 4-itemset*

--	--

- Dikarenakan *matrix* sudah *null* maka proses pencarian *frequent itemset* berhenti.
- Dari hasil akhir penggabungan maka dapat dibentuk aturan asosiasinya sebagaimana terdapat pada Tabel 12.

Tabel 12. *Generate Rule*

Kombinasi	Support Count	Confidence(%)
B17→B18∧B19	3	75
B18→B17∧B19	3	75
B19→B17∧B18	3	75
B17∧B18→B19	3	75
B17∧B19→B18	3	100
B18∧B19→B17	3	100
B17→B18∧B20	3	75
B18→B17∧B20	3	75
B20→B17∧B18	3	75
B17∧B18→B20	3	75
B17∧B20→B18	3	100
B18∧B20→B17	3	100
B18→B19∧B20	3	75
B19→B18∧B20	3	75
B20→B18∧B19	3	75
B18∧B19→B20	3	100
B18∧B20→B19	3	100
B19∧B20→B18	3	75
B17→B19∧B20	3	75
B19→B17∧B20	3	75
B20→B17∧B19	3	75
B17∧B19→B20	3	100
B17∧B20→B19	3	100
B19∧B20→B17	3	75

- Kemudian *rule* di seleksi sesuai dengan *minimum confidence* yang telah di tentukan.

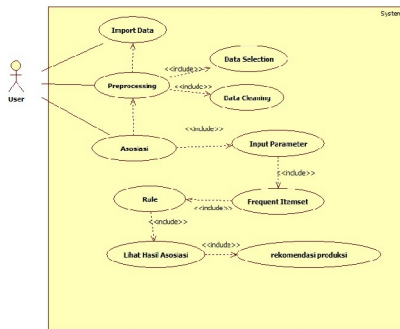
Tabel 13. Hasil seleksi *Rule*

Kombinasi	Support Count	Confidence(%)
B17∧B19→B18	3	100
B18∧B19→B17	3	100
B17∧B20→B18	3	100
B18∧B20→B17	3	100
B18∧B19→B20	3	100
B18∧B20→B19	3	100
B17∧B19→B20	3	100
B17∧B20→B19	3	100

3.4. Pemodelan Sistem

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem yang akan dibuat. Diagram *use case* yang terdapat pada

sistem yang akan dibangun terdiri dari satu *user* dan sepuluh *use case*.

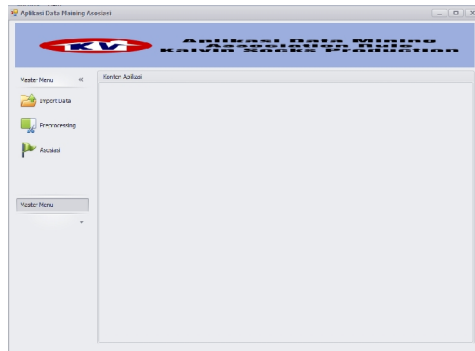


Gambar 1. Use Case Diagram

3.5 Implementasi Antarmuka

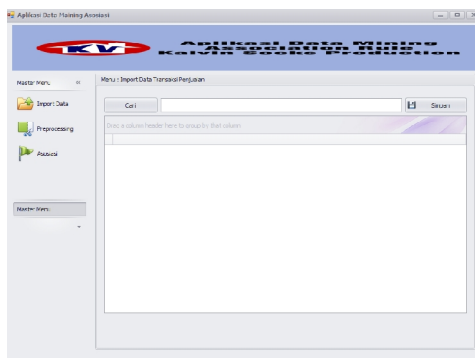
Adapun hasil implementasi dari pemodelan system diatas dapat dilihat dari tampilan dibawah ini.

a. Tampilan Halaman utama



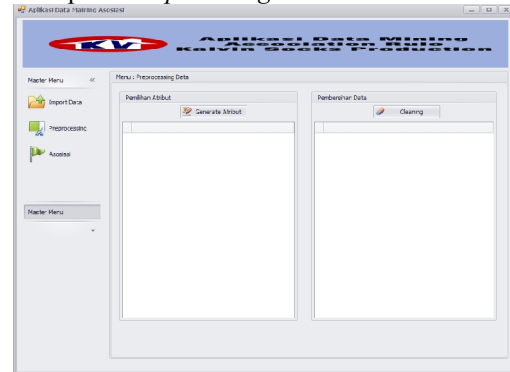
Gambar 2. Halaman Utama Aplikasi

b. Tampilan Import Data



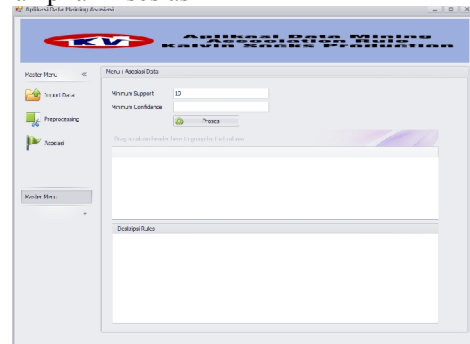
Gambar 3. Tampilan Menu Import Data

c. Tampilan Preprocessing



Gambar 4. Tampilan Menu Preprocessing

d. Tampilan Asosiasi

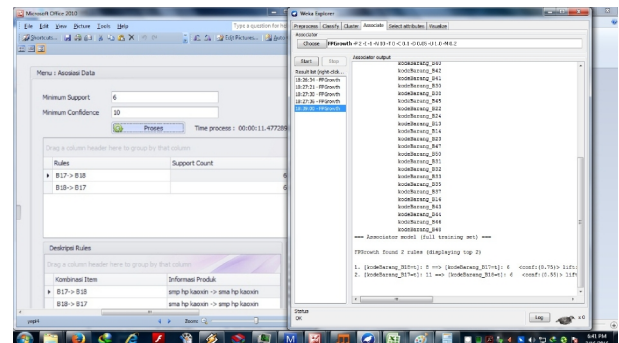


Gambar 5. Tampilan Menu Proses Asosiasi

4. Pengujian Penelitian

Adapun hasil penelitian diuji dengan menguji perangkat lunak yang telah dibangun apakah menghasilkan data yang *valid* dan sesuai dengan hasil penerapan metode *association rule* dengan algoritma *improved apriori* oleh karena itu, maka dilakukan uji banding hasil antara perangkat lunak yang dibangun dengan aplikasi *mining WEKA*

Pengujian menggunakan 27 data transaksi penjualan di bulan Agustus 2015 dengan nilai *minimum support* 6 dan nilai *minimum confidence* 10%. Aturan asosiasi yang dihasilkan adalah sebagaimana terlihat pada gambar 6.



Gambar 6. Perbandingan hasil rule aplikasi yang dibangun dengan aplikasi mining WEKA

Dari hasil pengujian antara aplikasi yang dibangun dengan aplikasi *mining* WEKA, *rule* yang didapat oleh aplikasi yang dibangun menggunakan algoritma *improved apriori* dan aplikasi WEKA adalah sama yaitu : B17 → B18 dan B18 → B17

Setelah pengujian perbandingan hasil *rule* antara aplikasi yang dibangun dengan aplikasi WEKA, maka pengujian selanjutnya yaitu membandingkan waktu dalam proses pencarian *rule* antara aplikasi yang dibangun dengan aplikasi yang sebelumnya telah digunakan oleh pihak KALVIN SOCKS PRODUCTION.

Pengujian menggunakan 178 data transaksi penjualan di bulan Agustus 2015 dengan nilai *minimum support* 10 dan nilai *minimum confidence* 60%. Aturan asosiasi yang dihasilkan sistem yang dibangun adalah seperti terlihat pada gambar 7.

Rule	Support Count	Confidence %
B17 -> B18	13	73%
B17 -> B18 -> B18	13	73%
B18 -> B17	13	73%
B18 -> B17 -> B17	13	73%

Gambar 7. Perbandingan hasil *rule* dan waktu antara aplikasi yang dibangun dengan aplikasi yang digunakan oleh KALVIN SOCKS PRODUCTION

Dari hasil pengujian antara aplikasi yang dibangun dengan aplikasi yang digunakan oleh KALVIN SOCKS PRODUCTION bahwa waktu yang dibutuhkan oleh aplikasi KALVIN SOCKS PRODUCTION yang menggunakan algoritma *apriori* dalam pencarian pola pembelian pelanggan membutuhkan waktu 15 menit 43 detik. Sedangkan aplikasi yang dibangun dengan algoritma *improved apriori* hanya membutuhkan waktu sekitar 1 menit 49 detik untuk mencari pola pembelian pelanggannya

Jadi, dapat disimpulkan bahwa algoritma *improved apriori* dapat meminimalkan masalah yang terjadi pada algoritma *apriori* dalam mengoptimisasi waktu proses pencarian pola pembelian pelanggan

5. PENUTUP

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan pada aplikasi yang menggunakan algoritma *improved apriori* dapat diambil kesimpulan sebagai berikut :

1. Aplikasi yang dibangun menggunakan algoritma *improved apriori* dapat meminimalkan masalah yang terjadi pada algoritma *apriori*
2. Aplikasi yang dibangun menggunakan algoritma *improved apriori* dapat mengoptimalkan waktu pemrosesan dalam pencarian pola pembelian.

DAFTAR PUSTAKA

- [1] Deasy Rusmawati, “Penerapan Data Mining Pada Penjualan Produk Kaos Kaki Di Pabrik Kalvin Socks Production Menggunakan Metode Association”, Skripsi, Teknik Informatika, Unikom, 2015
- [2] Vivul Mangla, Chandni Sarda, Sarthak Marda, “Improving The Efficiency Of Apriori Algorithm In Data Mining”, International Journal of Engineering and Innovative Technology (IJEIT), Volume III Issue III, 2013
- [3] Moh. Nazir, (2011), *Metode Penelitian*, (R. Sikumbang, Ed.), Bogor: Ghalia Indonesia.
- [4] Pressman, R. S., (2010), *Software Engineering A Practitioner’s Approach Seventh Edition*, United States: McGraw-Hill.
- [5] IBM Corp., (2011), *IBM SPSS Modeler CRISP-DM Guide*, USA.
- [6] Han & Kamber, (2006), *Data Mining Concept And Technique Second Edition*, San Fransisco: TheMorgan Kaufmann.
- [7] Krisdianto, N & Arymurthy, A.N., “Improved Apriori Berbasis Matrix Dengan Incremental Database Untuk Market Basket Analysis”, Prosiding Seminar Nasional Ilmu Komputer, Universitas Diponegoro, 2012.
- [8] Russ Miles & Kim Hamilton, (2006), *A Pragmatic Intoduction to UML, First Edition*, United States of America : O’reilly.
- [9] Dyer, R. J., (2008), *MYSQL IN A NUTSHELL*, Second Edition, United States of America : O’reilly.
- [10] Freeman, A., (2010), *Introduction Visual C# 2010*, United States of America : APRESS.

