

ANALISIS DAN IMPLEMENTASI ALGORITMA CT-APRIORI UNTUK ASOSIASI TRANSAKSI BARANG

ANALYSIS AND IMPLEMENTATION OF CT-APRIORI ALGORITHM FOR ASSOCIATION OF TRANSACTION

Septian Ardiansyah¹, Hetti Hidayati, S.Kom., M.T.², Amarilis Putri, S.T., M.T.³

^{1,2}Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom
septianosardiansvah@gmail.com¹, hettihd@telkomuniversity.ac.id²,
amarilisputri@telkomuniversity.ac.id³

Abstrak

Dalam *data mining*, aturan asosiasi digunakan untuk menemukan pola yang mendeskripsikan kekuatan di dalam data untuk mencari hubungan antar item[2]. Proses untuk menemukan hubungan antar item ini bisa melibatkan data yang banyak, maka diperlukan pembacaan data secara intensif, sehingga dibutuhkan waktu dan biaya komputasi yang besar. *Association Rules Mining* merupakan solusi untuk mengatasi permasalahan tersebut. Dengan *Association Rules Mining* dapat ditemukan informasi dari sekumpulan data dan menjadikannya suatu aturan asosiasi yang menggambarkan keterhubungan antar item[2][5]. Algoritma *CT-Apriori* merupakan revisi dari Algoritma *Apriori* yang sering digunakan dalam *Association Rule Mining*[10]. Pada penelitian ini akan mengimplementasikan algoritma *CT-Apriori* untuk menemukan aturan asosiasi dari data transaksi. Menggunakan dua nilai analisis penting yaitu *Minimum Support* dan *Minimum Confidence*[2][10]. Setiap *rules* yang didapatkan berasal dari data transaksi yang dilakukan oleh konsumen dalam berbelanja yang tersimpan di dalam *database*. Berdasarkan hasil pengujian, didapatkan *knowledge* berupa *rule* yang terdiri dari *item* yang dibeli dengan *item* yang dibeli bersamaan beserta nilai *support* dan *confidencenya*. Selain itu, didapatkan pula akurasi dari setiap *rules* yang didapatkan.

Kata kunci : data mining, association rule mining, CT-Apriori

Abstract

In data mining, association rules are used to find patterns that describe the strength in the data to find relationships between items[2]. The process to find the relationship between these items can involve a lot of data, it would require the reading of data intensive, so it takes time and a large computational cost. Association Rules Mining is a solution to overcome these problems. With the Association Rules Mining can be found information from a set of data and making an association rules that describe connectivity between items[2][5]. CT-Apriori algorithm is a revision of Apriori algorithms are often used in Association Rule Mining[10]. In this research will implement the CT-Apriori algorithm to discover association rules from transaction data. Using two important analytical values are Minimum Support and Minimum Confidence [2][10]. Any rules obtained derived from data transactions made by consumers. Based on the test results, the knowledge gained is the rules of association of the items purchased with the items purchased together with the value of its support and confidence. In addition, also obtained the accuracy of any rules that would be obtained.

Keywords : data mining, association rule mining, CT-Apriori

1. Pendahuluan

Banyak perusahaan retail yang mengumpulkan data, seperti data pembelian oleh *customer* yang dikoleksi setiap harinya. Dari data tersebut dapat digali informasi tambahan untuk membantu menentukan strategi usaha bagi perusahaan. Masalah inilah yang mendasari sehingga diperlukan sebuah kemudahan bagi pihak pengambil keputusan untuk memanfaatkan data yang tersimpan agar dapat digali beberapa informasi tambahan berdasarkan aktifitas transaksi, sehingga informasi yang didapat, dapat dijadikan rujukan oleh pihak pengambil keputusan dalam menentukan strategi penjualan berdasarkan informasi dari aktifitas belanja pelanggan dengan memanfaatkan *Association Rule Mining*.

Salah satu algoritma yang sering digunakan untuk menemukan aturan asosiasi adalah algoritma *Apriori*. Algoritma *Apriori* merupakan algoritma populer dalam pencarian *frequent itemset*[5][9]. Algoritma ini ditujukan untuk mencari kombinasi dari kumpulan item yang mempunyai suatu nilai keseringan tertentu dengan kriteria yang diinginkan *user*. Pada tugas akhir ini akan dibangun suatu sistem perangkat lunak dengan mengimplementasikan algoritma *Compact Transaction-Apriori (CT-Apriori)* untuk mencari pola asosiasi antar

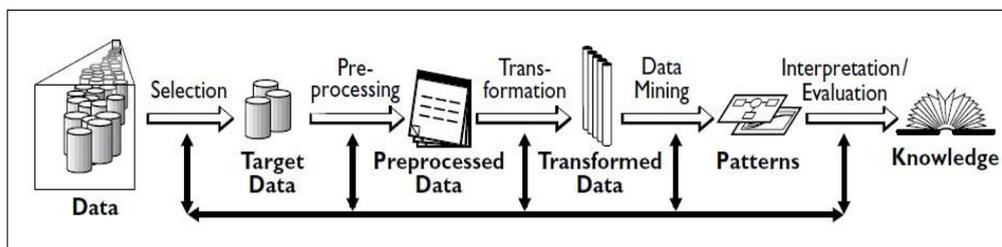
item yang memiliki keterkaitan ketika penjualan pada setiap transaksi. Algoritma *CT-Apriori* adalah pengembangan dari Algoritma *Apriori*[10]. Algoritma ini menggunakan data transaksi yang telah dikompres yang dinamakan *Compact Transaction Database (CTDB)*[10]. Untuk menghasilkan *CTDB* digunakan struktur data pohon untuk kompres data transaksi asli yang dinamakan *Compact Transaction-Tree (CT-Tree)*[10]. *CTDB* yang dihasilkan memiliki dua bagian, yaitu *head* dan *body*. *Head* dari *CTDB* berisi jumlah kemunculan setiap item dari transaksi dan *body* dari *CTDB* berisi jumlah transaksi yang terjadi, pembentukan *frequent 1-itemset* didapat dari *head CTDB* sehingga lebih efisien dengan dengan melewati *scan* awal pada *database* untuk melakukan *frequent pattern mining*.

Diharapkan dengan diimplementasikannya algoritma *CT-Apriori* terhadap transaksi penjualan barang maka dapat dijadikan pengetahuan baru berupa rekomendasi keterkaitan antar item bagi *market analyst* yang dapat berguna untuk strategi dalam penjualan maupun mengatur stok pada produk.

2. Dasar Teori

2.1 Konsep Data Mining

Data mining merupakan bagian dari *Knowledge Discovery in Databases (KDD)*[2]. *KDD* adalah keseluruhan proses pengolahan data yang belum diproses untuk mendapatkan pengetahuan yang bermanfaat[2][5]. Istilah *data mining* dan *knowledge discovery in databases (KDD)* sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data. Namun sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi memiliki keterkaitan satu sama lain. Salah satu tahapan dalam keseluruhan proses *KDD* adalah *data mining*[5]. Proses *KDD* secara garis besar digambarkan sebagai berikut:



Gambar 2.1 Proses Knowled Discovery in Databases[2]

Proses *KDD* secara garis besar terdiri dari lima tahap. Namun, dalam proses *KDD* yang sesungguhnya, dapat saja terjadi iterasi atau pengulangan pada tahap tahap tertentu. Pada setiap tahap dalam proses *KDD*, seorang analis dapat saja kembali ke tahap sebelumnya. Sebagai contoh, pada saat coding atau data *mining*, analis menyadari proses *cleaning* belum dilakukan dengan sempurna, atau mungkin saja analis menemukan data atau informasi baru untuk “memperkaya” data yang sudah ada.

2.2 Association Rule Mining

Association rule mining merupakan teknik *data mining* untuk menemukan aturan asosiasi antar kombinasi *item* yang ada pada sekumpulan data. Dalam pencarian *Association rule*, diperlukan suatu variabel ukuran yang dapat ditentukan oleh user, yaitu nilai *Support* dan *Confidence* untuk mengatur batasan sejauh mana dan sebanyak apa hasil *output* yang diinginkan oleh user. *Support* merupakan persentase transaksi yang dipilih secara random dari sekumpulan data[5].

Nilai *support 1-itemset* dan nilai *2-itemset* dapat diperoleh dari persamaan berikut berikut :

$$Support(X) = \frac{count(X)}{count(D)} \tag{2.1}$$

$$Support(X \rightarrow Y) = \frac{count(X \cup Y)}{count(D)} \tag{2.2}$$

Sedangkan *Confidence* merupakan persentase kuatnya hubungan antar *item*[5].

$$Confidence(X \rightarrow Y) = P(Y|X) = \frac{count(X \cup Y)}{count(X)} \tag{2.3}$$

Secara umum *Association rule mining* dapat dikatakan sebagai proses dua tahap:

1. Cari semua *itemset* yang sering muncul; *itemset* yang dimaksud memenuhi *minimum support*.
2. Hasilkan *Association rule* yang strong dari *itemset*; *rule* ini harus memenuhi *minimum support* dan *minimum confidence*.

2.3 Algoritma CT-Apriori

Algoritma *CT-Apriori* merupakan pengembangan dari algoritma *Apriori* yang merupakan algoritma populer dalam *Association Rules Mining* untuk pencarian *frequent itemset*. Algoritma *CT-Apriori* menghasilkan *frequent itemset* dari database transaksional asli (*TDB*) yang telah dikompres yang dinamakan *Compact Transaction Database (CTDB)*[10]. *CTDB* dihasilkan dengan menggunakan struktur data pohon yang dinamakan *Compact Transaction-Tree (CT-Tree)*, yang akan melakukan *scanning* data untuk menghasilkan *Compact Transaction Database* dari database transaksi asli[10]. *CTDB yang terbentuk* merupakan database yang memiliki dua bagian, yaitu *Head* dan *Body*. Bagian *head* dari *CTDB* berisi jumlah kemunculan setiap *item* dari *TDB* dan bagian *body* berisi transaksi unik yang terjadi.

Dalam membangun *CT-Tree* setiap simpul dari *tree* memiliki dua *record* yaitu *V* dan *Vc*. *V* merupakan *item* di *TDB* dan *Vc* adalah jumlah terjadinya transaksi di *TDB* yang terdiri dari semua *item* dan digambarkan di cabang *tree* dari akar sampai ke simpul *V*[10]. Diberikan data transaksi sebagai berikut :

Tabel 2.1 Dataset transaksi

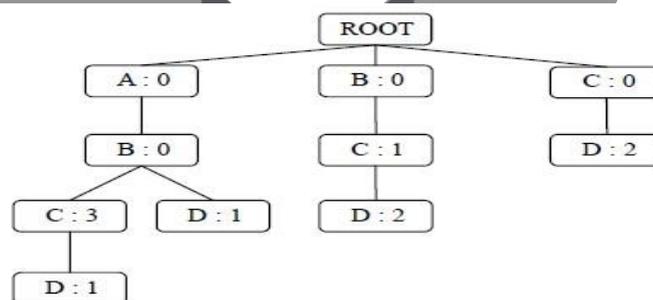
TID	List Item Id
001	A,B,C,D
002	A,B,C
003	A,B,D
004	B,C,D
005	A,B,C
006	A,B,C
007	B,C
008	C,D
009	B,C,D
010	C,D

Untuk transaksi pertama dari dataset di atas, transaksi pertama {A,B,C,D} membentuk cabang pertama *Tree* yang dibangun sebagai berikut: {(A:0), (B:0), (C:0), (D:1)}. Node terakhir (D:1) mencatat terjadinya jalur ABCD. Pada saat yang sama, jumlah kemunculan *itemnya* dicatat dalam daftar sebagai [A:1, B:1, C:1, D:1].

Untuk transaksi kedua {A,B,C} berbagi jalur {A,B,C} pada cabang pertama, sehingga tidak perlu dibuat cabang baru, namun jumlah kejadian yang terakhir pada simpul bersama nilainya dihitung satu sebagai (C:1). Dan jumlah kemunculan setiap *item* dalam transaksi ini bertambah satu pada daftar, yaitu [A:2, B:2, C:2, D:1].

Untuk transaksi ketiga {A,B,D} berbagi jalur {A,B} pada cabang pertama, sehingga dibuat satu node baru (D:1) dan dihubungkan sebagai child dari (B:0) dan daftar jumlah kemunculan *itemnya* menjadi [A:3, B:3, C:2, D:2].

Transaksi keempat dan kelima mengarah ke pembangunan dua cabang *tree*, masing-masing {(B:0), (C:0), (D:1)} dan {(C:0), (D:1)}. Dan daftar jumlah frekuensi menjadi [A:3, B:4, C:4, D:4].



Gambar 2.2 CT-Tree yang Terbentuk[10]

Setelah pembacaan semua transaksi, *CT-Tree* lengkap dari *TDB* ditunjukkan pada Gambar 2.2. Jumlah kemunculan *itemnya* adalah [A:5, B:8, C:9, D:6].

Kemudian jumlah kemunculan tiap *item* diurutkan secara menurun sehingga menjadi {C:9, B:8, D:6, A:5} dan ditulis sebagai *Head* dari *CTDB*. Sedangkan bagian *body* dari *CTDB* dibentuk berdasarkan simpul dari setiap *item* yang nilainya lebih besar dari 0 di *CT-Tree* kemudian dicatat dan dihitung sebagai transaksi unik yang terjadi. Urutan *item* diurutkan secara menurun dan dicatat sebagai transaksi unik yang terjadi. Keempat *item* dari *TDB* {A,B,C,D} tercatat di *Head* dengan mencatat kemunculan setiap *item* yang diurutkan secara menurun berdasarkan kemunculannya dari yang tertinggi hingga terendah. Sedangkan bagian *Body* dari *CTDB* berisi transaksi unik yang terjadi, transaksi dengan *item* {C,B,A} terjadi tiga kali, sedangkan transaksi dengan *item*

{C,B,D} dan {C,B} masing-masing terjadi dua kali, sehingga transaksi unik yang terjadi ada enam transaksi dari sepuluh transaksi yang merupakan total keseluruhan transaksi.

Tabel 2.2 Compact Transaction Database

Head				
Item	C	B	D	D
Count	9	8	6	5
Body				
Count	List of ItemIDs			
3	C,B,A			
1	C,B,D,A			
1	B,D,A			
1	C,B			
2	C,B,D			
2	C,D			

Pengurutan *item* berdasarkan jumlah kemunculan masing-masing *item* di *head* dari *CTDB* yang memenuhi minimum support. Algoritma *CT-Apriori* melewati scan awal database yang dilakukan algoritma *Apriori* dan menentukan *frequent 1-itemset* dengan membaca *head* dari *CTDB*. Kemudian calon *2-itemset* dihasilkan dari *F1* dan menghitung terjadinya transaksi pada *body* dari *CTDB* sampai semua *frequent itemset* terbentuk.

2.4 Pembangkitan Rule Asosiasi

Rule asosiasi dibangkitkan dari *frequent itemset* yang dihasilkan pada proses sebelumnya. *Rule* asosiasi dikatakan kuat jika *confidence*-nya lebih besar atau sama dengan *minconf*. Nilai *confidence* suatu *rule* asosiasi menyatakan kekuatan prediksi dari *rule* tersebut.

2.5 Menghitung Akurasi

Setelah didapatkan *rule* kemudian dievaluasi seberapa prediktif model yang sudah terbentuk. Dilakukan proses evaluasi, semua data (*training + testing*) digunakan untuk menentukan akurasi. Pengujian akurasi *rule* yang terbentuk yaitu dengan membandingkan *rule* hasil proses data mining pada data *training* dengan data *testing*. [5]. Penghitungan akurasi *association rule* dilakukan dengan menggunakan persamaan.

$$\frac{\text{Jumlah Data yang Sesuai}}{\text{Jumlah Data}} = 100\% \tag{2.4}$$

Akurasi = 100% - error.

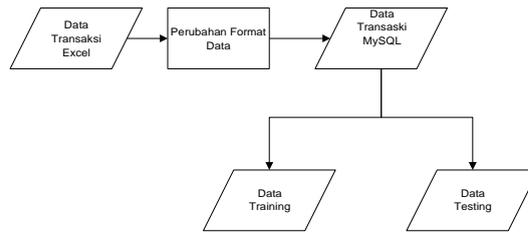
3. Pembahasan

3.1 Deskripsi Umum Sistem

Tujuan dari tugas akhir ini adalah mengimplementasikan algoritma *CT-Apriori* ke dalam suatu perangkat lunak dengan memanfaatkan data transaksi untuk mencari aturan asosiasi. Aturan asosiasi yang didapat berupa informasi item yang dibeli bersamaan dengan nilai *support* dan *confidence* beserta akurasinya. Tahapan analisis sistem adalah tahapan mengidentifikasi kebutuhan fungsional dalam mempersiapkan rancang bangun implementasi yang bertujuan untuk merancang dan mendesain sistem dalam memenuhi kebutuhan pemakai sistem. Untuk itu dilakukan proses dimulai dari pencarian kebutuhan data transaksi sampai mencari *rule* yang akan dijadikan bahan evaluasi bagi *market analyst* sebagai bahan pertimbangan untuk menentukan keputusan yang tepat.

3.2 Rancangan Sistem

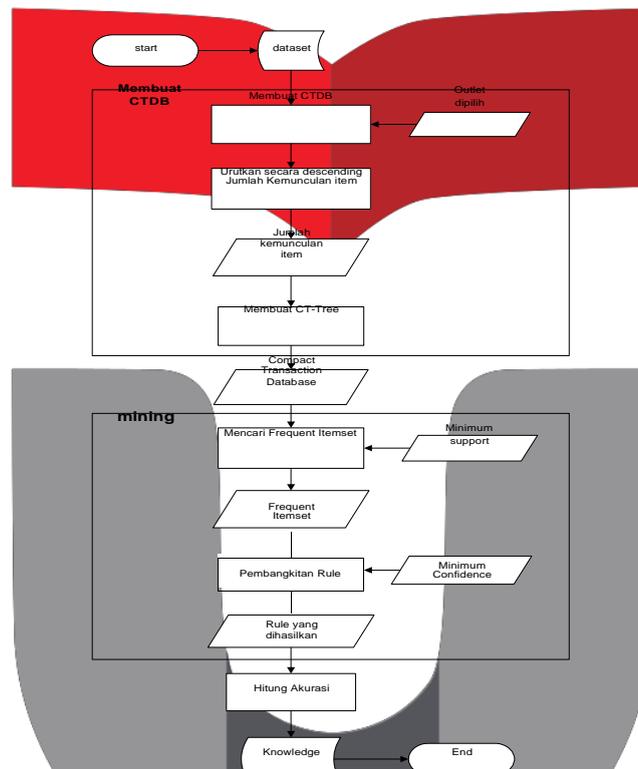
Tahap pertama dimulai dari *data preprocessing*. Dalam tahap ini, dilakukan, data yang diperoleh mengalami perubahan format data. dari format *XLS* ke *CSV* untuk di impor ke dalam *database MySQL*. Kemudian data transaksi dibagi menjadi dua bagian, data *training*, yaitu dataset yang digunakan untuk pembangkitan *rule* dan data *testing*, yaitu dataset yang digunakan untuk pengujian terhadap *rule*. Dataset pembangkitan *rule* adalah dataset yang akan digunakan dalam proses pembangkitan *frequent itemset* sampai dengan munculnya hasil output sistem yaitu berupa *rule* berbentuk aturan asosiasi. Sedangkan, data pengujian terhadap *rule* adalah dataset transaksi asli yang akan digunakan untuk menguji *rule* yang telah dihasilkan oleh sistem. Seleksi data adalah pemilihan atribut mana saja dari atribut data yang akan digunakan sehingga data tersebut dapat diolah sesuai dengan kebutuhan proses *data mining*. Pemilihan atribut data yang akan digunakan dalam penelitian ini dilakukan dengan manual menggunakan perintah *SELECT* pada *MySQL*.



Gambar 3.1 Preprocessing Data

Pembangunan *CTDB* menjadi tahap awal sebelum sistem menghasilkan output *Association rules*. Proses *mining* dengan Algoritma *CT-Apriori* menghasilkan kumpulan *frequent itemset* dengan *minimum support* yang ditentukan pengguna, kemudian dibangkitkan *rule*-nya dan dihitung masing-masing nilai *confidence*-nya.

Sistem mengeluarkan hasil dari penghitungan *confidence* yang akan dijadikan sebagai *rule*. Akurasi dari *rule* didapatkan dari hasil uji dengan data testing dan menghasilkan knowledge yang dapat dijadikan acuan dalam menganalisis asosiasi ataupun pola keterkaitan antar *item*.



Gambar 3.2 Gambaran Umum Sistem

3.3 Analisis Kebutuhan Sistem

3.3.1 Kebutuhan Fungsionalitas

Adapun kebutuhan fungsionalitas sistem yang dibangun adalah sebagai berikut.

- Sistem dapat membangun *CTDB* yang akan digunakan bersamaan dengan algoritma *CT-Apriori*. dalam membangkitkan *rule* asosiasi
- Sistem dapat memasukkan nilai *minimum support*
- Sistem dapat memasukkan nilai *minimum confidence*
- Sistem dapat menghitung nilai *support* dan *confidence* tiap *itemset*
- Sistem dapat menemukan aturan asosiatif antara suatu kombinasi *item*.
- Sistem dapat menampilkan informasi data hasil pencarian aturan asosiatif berupa *item* yang dibeli dan *item* yang dibeli pula.
- Proses *testing* dengan melakukan pengujian dataset *testing* terhadap *rule* yang dihasilkan untuk mengetahui akurasinya.

3.3.2 Spesifikasi Perangkat Keras

Sistem diimplementasikan pada spesifikasi perangkat keras, yaitu dengan rincian

- Processor* Intel(R) Core(TM) i5-3210M 2.50GHz
- Memori 4 GB RAM
- Monitor 17"

3.3.3 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem sebagai berikut :

- Sistem operasi : Windows 7 Ultimate 32-bit
- Database : MySQL Client (XAMPP 1.7.3)
- Bahasa Pemrograman : Java
- Tools pemrograman : Netbeans IDE

4. Pengujian dan Analisis

4.1 Dataset

Tabel 4.1 Dataset

No	Outlet	Jumlah Data	Jumlah Transaksi
1.	F03	2041	1313
2.	F04	2021	1260
3.	F07	2133	941
4.	F08	2099	935
Total		8294	4449

Masukan dari sistem berupa data transaksi penjualan butik busana muslim Siti Hajar Tren Islami yang terdiri dari 4 outlet. Data tersebut dibagi menjadi dua bagian yakni *dataset training*, yaitu *dataset* untuk pembangkitan *rule* dan *dataset testing*, untuk pengujian terhadap *rule* yang dihasilkan. Hal ini dilakukan untuk mendapatkan akurasi dari setiap *rule* yang dihasilkan

Tabel 4.2 Dataset Pembangkitan Rule

No.	Outlet	Waktu	Jumlah Data	Jumlah Transaksi
1.	F03	1 Juli 2014-23 September 2014	1104	698
2.	F04	3 November 2014-17 Maret 2015	1084	646
3.	F07	11 September 2014-2 Desember 2014	1109	465
4.	F08	12 September 2014-31 Desember 2014	1002	464
Total			4299	2273

Tabel 4.3 Dataset Uji Rule yang Dihasilkan

No	Outlet	Waktu	Jumlah Data	Jumlah Transaksi
1.	F03	23 September 2014 – 11 November 2014	1038	615
2.	F04	17 Maret 2015 – 14 Mei 2015	1012	614
3.	F07	4 Desember 2014 – 7 April 2015	1109	474
4.	F08	2 Januari 2015 – 8 Mei 2015	1101	473
Total			4260	2176

4.2 Pengujian

Pengujian dilakukan dengan dataset pada tabel 4.2, semua outlet dilakukan pengujian dengan nilai minimum support antara 0.1% s/d 0.5% dengan minimum confidence 10% s/d 50%. Pemilihan nilai minimum support dan minimum confidence ini dimaksudkan untuk memaksimalkan pencarian frequent itemset dan pembentukan rule asosiasi dan melihat perbedaan antara pengujian dengan minimum support dan minimum confidence satu dengan yang lain sehingga dapat dilihat nilai optimum untuk setiap outlet.

Frequent itemsets yang dihasilkan akan dibangkitkan rule-nya. Rule yang dihasilkan akan dibandingkan dengan dataset pada tabel 4.3 untuk mendapatkan akurasi dari rule yang didapatkan.

Tujuan dengan skenario pengujian, antara lain sebagai berikut:

- Menganalisis pengaruh nilai *minimum support* terhadap jumlah *frequent itemset* yang dihasilkan.
- Menganalisis pengaruh nilai *minimum confidence* terhadap *rule* yang dihasilkan dari *frequent itemset* dan waktu prosesnya.
- Menganalisis pengaruh nilai *minimum support* dan *minimum confidence* terhadap akurasi dari *rule* yang dihasilkan.

4.3 Hasil Pengujian

Dari hasil pengujian pengaruh nilai support terhadap frequent itemset pengaruh nilai minimum confidence terhadap rule yang dihasilkan dari frequent itemset dan waktu prosesnya dari tiap outlet diperoleh hasil sebagai berikut :

Tabel 4.4 Jumlah Frequent Itemset Semua Outlet

Outlet	Support (%)				
	0.1	0.2	0.3	0.4	0.5
F03	55	39	32	21	21
F04	86	57	39	39	16
F07	46	46	46	22	22
F08	48	48	24	24	22

Dari hasil pengujian pengaruh nilai support terhadap pengaruh nilai minimum confidence terhadap rule yang dihasilkan dari frequent itemset dan waktu prosesnya dari tiap outlet diperoleh hasil sebagai berikut :

Tabel 4.5 Pengaruh Nilai Support dan Confidence Terhadap Jumlah Rule Outlet F07

Confidence	Support(%)	Jumlah rule	Waktu(ms)
10	0.1	12	23841
	0.2	12	23635
	0.3	9	22954
	0.4	9	22788
	0.5	9	22615
20	0.1	11	22787
	0.2	11	22635
	0.3	8	22934
	0.4	8	22154
	0.5	8	22105
30	0.1	11	21639
	0.2	11	21438
	0.3	8	21054
	0.4	8	20975
	0.5	8	19882
40	0.1	8	21418
	0.2	8	20986
	0.3	7	19874
	0.4	7	19385
	0.5	7	19132
50	0.1	7	21087
	0.2	7	19867
	0.3	6	19654
	0.4	6	19005

Dari hasil pengujian pengaruh nilai support terhadap frequent itemset pengaruh nilai minimum confidence terhadap rule yang dihasilkan dari frequent itemset dan waktu prosesnya dari tiap outlet diperoleh hasil sebagai berikut :

Tabel 4.6 Pengaruh Nilai Support dan Confidence Terhadap Akurasi Outlet F07 (%)

Support (%)	Confidence (%)				
	10	20	30	40	50
0.1	31.94	32.58	32.58	40.62	46.63
0.2	31.94	32.58	32.58	40.62	46.63

Tabel 4.6 lanjutan

0.3	38.89	40.62	40.62	46.63	54.17
0.4	38.89	40.62	40.62	46.63	54.17
0.5	38.89	40.62	40.62	46.63	54.17

5.1 Kesimpulan

Nilai *support* mempengaruhi jumlah *frequent itemset*, *rule* yang dihasilkan dan lama waktu eksekusi program. Semakin besar nilai *support* maka akan semakin sedikit jumlah *frequent itemset* yang dihasilkan sehingga *rule* yang didapat semakin sedikit dan waktu pemrosesan menjadi lebih cepat. Nilai *confidence* mempengaruhi jumlah *rule* yang dihasilkan dan lama waktu eksekusi program. Semakin besar nilai *confidence* maka akan semakin tinggi batas pencarian terhadap jumlah pola keterkaitan antar produk yang dibeli secara bersama-sama yang dimasukkan kedalam perhitungan sistem mempengaruhi waktu eksekusi menjadi lebih cepat. Nilai *support* dan *confidence* tidak mempengaruhi akurasi dari *rule* yang dihasilkan hal ini adalah pengaruh dari penentuan komposisi data training dan data testing. Pada pengujian ini *frequent itemset* tertinggi dihasilkan pada *minimum support* 0.1% dan jumlah *rule* paling dihasilkan dari *frequent itemset* yang didapat pada *minimum confidence* 10% namun dengan waktu proses yang lebih lama sehingga nilai optimum akurasi berdasarkan akurasi dan waktu proses yang didapat pada masing-masing outlet berbeda-beda.

5.2 Saran

Ada baiknya untuk penelitian selanjutnya algoritma *CT-Apriori* dibandingkan dengan algoritma lain yang mendukung *association rule* sehingga dapat dibandingkan algoritma mana yang lebih efektif dan efisien dalam menyelesaikan permasalahan *association rule mining*. Diharapkan adanya pengujian sistem menggunakan data lain sehingga bisa dilihat perbedaan performansi algoritma ini terhadap jenis data lain. Hasil dari sistem ini hanya sebatas melihat keterhubungan antara 2 item atau lebih yang sering dibeli secara bersamaan oleh pelanggan dalam satu kali transaksi pembelian. Untuk kedepan diharapkan adanya penambahan parameter uji sehingga sistem dapat menghasilkan output berupa *association rule* yang lebih bermanfaat dan berguna sebagai rekomendasi terhadap pengambil keputusan

DAFTAR PUSTAKA

- [1] A.B. Dhivya, B. Kalpana. 2010. *A Study on the Performance of CT-APRIORI and CT-PRO Algorithms using Compressed Structures for Pattern Mining*. Journal of Global Research in Computer Science.
- [2] H. David, M. Heikki and S. Padhraic. *Principles of Data Mining* vol. 49, no. 2, pp. 77-82, 2001.
- [3] H. Andreas, S. Budhi, R. Hendra. 2008. *Aplikasi Data Mining untuk meneliti Asosiasi Pembelian Item Barang di Supermarket dengan Metode Market Basket Analysis*. Universitas Petra, Surabaya.
- [4] J. Han, J. Pei, Y. Yin, and R. Mao. 2004. *Mining Frequent Patterns without Candidate Generation: A Frequent pattern Tree Approach*. Data Mining and Knowledge Discovery: An International Journal, Kluwer Academic Publishers.
- [5] J. Han, M. K. Harcourt. 2006. *Data Mining Concepts and Techniques*.
- [6] Nugroho, Adi. 2009. *Rekayasa perangkat lunak menggunakan UML dan Java*. Yogyakarta : Andi.
- [7] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. 2000. *A Tree Projection Algorithm for Generation of Frequent itemsets*. Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining).
- [8] R. Agarwal, R. Srikant. 1998. *Fast Algorithms for Mining Association Rules*. IBM Almaden Research Center.
- [9] Ruldeviyani, Yova, Muhammad Fahrian. 2008. *Implementasi Algoritma-Asosiasi Rules Sebagai Bagian dari Pengembangan Data Mining Algorithms Collections*. Universitas Indonesia.
- [10] Qian Wan and Aijun An. 2005. *Compact Transaction Database For Efficient Frequent Pattern Mining*. York University.