

PENERAPAN METODE MATRIX AND INTERESTINGNESS BASED ASSOCIATION RULE MINING (MibARM) UNTUK REKOMENDASI PEMBELIAN PADA WEB COMMERCE

Lia Setiawan, 0910962008

Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA Universitas Brawijaya Malang

Abstrak

Pada saat ini, terdapat banyak sekali toko-toko virtual yang menjual beranekaragam barang dan jasa, sehingga pembeli tidak perlu keluar rumah untuk membeli sesuatu. *Web Commerce* merupakan toko virtual yang bisa digunakan untuk melakukan pembelian atau penjualan secara langsung.

Untuk meningkatkan penjualan di *web Commerce* salah satu cara adalah memberikan rekomendasi pembelian barang kepada pembeli. Metode yang sering digunakan untuk pembentukan rekomendasi adalah metode Association Rule Mining dengan Algoritma Apriori. Untuk meningkatkan kinerja dari Apriori maka diterapkan metode Matrix And Interestingness Based Association Rule Mining (MibARM) yang digunakan dalam penelitian ini.

Dari hasil uji coba metode MibARM pada 500 data transaksi, didapatkan *rule* yang memiliki rata-rata nilai Confidence 0.67, rata-rata nilai Interestingness 7.57 dan rata-rata nilai Lift 2.1. Sedangkan dari hasil uji coba pada 1000 data transaksi didapatkan rata-rata nilai Confidence 0.67, rata-rata nilai Interestingness 15.54 dan rata-rata nilai Lift Ratio 2.67.

Kata Kunci: rekomendasi pembelian, E-Commerce, Association Rule Mining, MibARM.

1. Pendahuluan

Di Indonesia pada saat ini terjadi perkembangan toko virtual yang cukup pesat, berbagai jenis *web Commerce* bermunculan menawarkan beranekaragam barang dan jasa kepada konsumen. Strategi bisnis seperti ini penyedia dan penjual tidak perlu memiliki modal yang besar dan berbagai persyaratan rumit yang dibutuhkan untuk mendirikan sebuah toko virtual melalui *web Commerce*, kemudahan ini memudahkan setiap orang

bisa memiliki satu atau banyak toko virtual yang digunakan untuk menawarkan produk atau jasa mereka melalui internet.

Electronic Commerce (E-Commerce) adalah penjualan dan pembelian barang atau jasa melalui media internet^[1]. Salah satu layanan dalam *web Commerce* untuk meningkatkan penjualan produk adalah sistem rekomendasi, sistem rekomendasi digunakan oleh *web Commerce* untuk menyarankan produk-produk kepada pelanggan mereka dan menyediakan informasi bagi pelanggan untuk membantu mereka memilih produk-produk yang akan dibeli. Produk-produk ini bisa direkomendasikan berdasarkan penjualan terbanyak pada situs tersebut atau berdasarkan analisis dari perilaku pembelian pelanggan sebagai prediksi untuk perilaku pembelian yang akan datang.

Metode Association Rule Mining adalah salah satu metode yang digunakan untuk rekomendasi pembelian ini. Hasil dari rekomendasi ini bergantung pada kualitas dan kuantitas dari *association rule* yang dibentuk. Pada umumnya algoritma yang dipakai dalam metode Association Rule Mining ini adalah algoritma Apriori yang diusulkan oleh Agrawal tahun 1993, dimana telah diaplikasikan di beberapa tipe sistem rekomendasi pembelian seperti *web Amazon* dan *eBay*.

Dalam penerapannya algoritma Apriori ini mempunyai beberapa kekurangan, diantaranya^[2]:

1. Algoritma ini tidak efisien karena dalam proses pencarian *frequent pattern*, dibutuhkan berulang kali pemindaian *database* sebanyak jumlah kandidat Itemsets C_k untuk membentuk *frequent itemsets* L_k dari kandidat item C_k . Sementara itu, metode ini menghasilkan banyak kandidat *itemset* yang berlebih atau mubazir, ini menyebabkan layanan rekomendasi tidak maksimal.
2. Algoritma ini membutuhkan waktu pemrosesan yang lebih lama, karena banyak terjadi perulangan *rule* ketika proses

pembentukan *rule*, sehingga menyebabkan penurunan kinerja rekomendasi tersebut.

Untuk meningkatkan efisiensi dari algoritma Apriori ini berbagai peneliti sebelumnya telah mengajukan berbagai jenis algoritma seperti yang dilakukan oleh T.J Tsay yang mengajukan metode FIUT yang melakukan efisiensi pada proses pembentukan *frequent itemset*^[3], Song juga mengajukan algoritma Index-BitTableFI untuk menghindari perulangan pada proses pengecekan *frequent itemsets*^[4]. Deng sendiri mengajukan algoritma Matrix and Interestingness based Association Rule Mining (MibARM) yang hanya sekali melakukan pemindaian *database* dan menggunakan matriks dalam pembentukan *frequent itemset*^[5].

Dari hasil pengujian yang dilakukan Deng diantara algoritma Apriori, CBAR, BitTableFI dan MibARM menunjukkan bahwa MibARM menurunkan terjadinya perulangan kandidat *itemset* dan menurunkan jumlah perulangan *rule* sehingga penulis mengimplementasikan algoritma MibARM dalam penyusunan jurnal ini, sesuai pada jurnal yang disusun oleh Deng dkk berjudul "An Efficient Association Rule Mining Method for Personalized Recommendation in Mobile E-Commerce" dengan judul jurnal "Penerapan Metode Matrix and Interestingness based Association Rule Mining (MibARM) untuk Rekomendasi Pembelian Pada web Commerce".

2. Definisi MibARM

Untuk meningkatkan kinerja metode Association Rule dan meningkatkan kualitas rekomendasi maka algoritma MibARM diterapkan. Algoritma ini hanya sekali melakukan *database scanning* kemudian membuang *non-frequent items* dalam proses *mining* untuk kompresi ruang pencarian dan menghindari perulangan kandidat *itemset*. Algoritma MibARM dalam prosesnya mempunyai dua tahap yaitu: *frequent itemset mining* dan *association rule generation*.

2.1 Frequent Itemset Mining berdasarkan Transaction Matrix

Untuk menghindari terjadinya *database scanning* secara berulang-ulang dalam pembentukan kandidat *itemset* maka

digunakan Transaction Matrix (*T*) untuk menyimpan data transaksi, dimana dalam prosesnya metode MibARM hanya melakukan *database scanning* sekali yang disimpan dalam sebuah Transaction Matrix (*T*) kemudian menghilangkan *non-frequent items* dalam proses *mining* untuk menghemat ruang pencarian dan untuk menghindari pengulangan kandidat *itemset*. Algoritma MibARM ini telah ditingkatkan untuk menangani *frequent k-Itemset* ($k > 2$). Dalam pembentukan *frequent itemset* terdapat beberapa tahapan, antara lain (Deng, 2010):

Definisi.1 Transaction Matrix (T)

Proses pertama ini merubah data transaksi dalam *database* menjadi bentuk Transaction Matrix (*T*) dua dimensi yang akan digunakan dalam proses *mining* selanjutnya. Pembentukan matriks ini ditunjukkan pada persamaan 1, dimana elemen baris dalam matriks transaksi adalah transaksi-transaksi dalam *database* (*D*) dan $I = \{I_j | j = 1, 2, \dots, n\}$ adalah kumpulan *n item* yang berbeda dalam *database*. Jika baris (*i-th*) transaksi (*T_i*) termasuk pada *item* (*I_j*), maka elemen pada *matrix* $T(t_{ij}) = 1$, jika tidak maka elemen $t_{ij} = 0$.

$$T = (T_1, T_2, \dots, T_m)' = (t_{ij})_{m \times n} \quad (1)$$

Dimana:

T = Transaction Matrix

$T_1, T_2, \dots, T_m$ = transaksi yang berbeda pada *database*

t_{ij} = entry yang terjadi pada inteseksi di baris ke (*i*) dan kolom ke (*j*) pada matriks *T*

m = jumlah baris

n = jumlah kolom

Definisi.2 Absolute Support (SupA)

Absolute Support (*SupA*) adalah jumlah kemunculan *item* (*I_j*) dalam matriks *T*. Setelah Transaction Matrix (*T*) terbentuk, langkah selanjutnya adalah menghitung semua Absolute Support dengan menjumlahkan semua element dari kolom ke *j* (*j-th*) pada matriks *T*, penghitungan Absolute Support ditunjukkan pada persamaan 2.

$$SupA(I_j) = \sum_{i=1}^m t_{ij}, i = 1, 2, \dots, m \quad (2)$$

Dimana:

$SupA$ = Absolute Support atau jumlah kemunculan *item*

$SupA(I_j)$ = Absolute Support dari *item* I_j

I_j = kumpulan *item* yang berbeda dalam database $j=1, 2, \dots, n$

Definisi.3 Support (Sup)

Setelah didapat $SupA$ kemudian dihitung juga Support (Sup) setiap *item* (i_j) dalam matriks T . Support dari sebuah *item* adalah $SupA$ dibagi dengan jumlah semua transaksi m . Untuk menghitung Sup setiap *item* dapat dilihat pada persamaan 3.

$$Sup(I_j) = \frac{SupA(I_j)}{m} = \frac{\sum_{i=1}^m t_{ij}}{m} \quad (3)$$

Dimana:

Sup = Support

$Sup(I_j)$ = Support dari *item* (I_j)

$SupA(I_j)$ = Absolute Support *item* (I_j)

m = jumlah transaksi

Definisi.4 Mining Frequent 1-Itemset

Frequent 1-itemset (L_1) didapatkan dari proses pencarian kandidat *Frequent 1-Itemset* (C_1) dengan menghilangkan kolom dari *item* (i_j) pada T yang memiliki Support (Sup) yang kurang dari minimal Support (Sup_{min}) yang ditetapkan. Persamaan dari pencarian L_1 dapat dilihat pada persamaan 4.

$$L_1 = \{I_j | Sup(I_j) \geq Sup_{min}, j = 1, 2, \dots, n\} \quad (4)$$

Dimana:

L_1 = *Frequent 1-itemset*

I_j = *item* dalam matriks T

Sup = Support *item*

Sup_{min} = minimal Support

Setelah dihilangkan kolom termasuk elemen *item* (I_j) yang mempunyai Support yang kurang dari Support minimal maka didapatkan *frequent 1-itemset* (L_1).

Langkah selanjutnya adalah menghilangkan elemen baris dalam matriks T yang jumlahnya kurang dari 2 ($\sum_{j=1}^n t_{ij} < 2$).

Jumlah baris dinotasikan m_2 dan kolom dengan n_2 ($m_2 \leq m$, $n_2 \leq n$).

Definisi.4 Mining Frequent 2-Itemsets

Untuk mencari *frequent 2-itemset* (C_2) yang pertama adalah membentuk sebuah Hermite matrix (H) dari matriks T . Persamaan ini ditunjukkan pada persamaan 5.

$$H = T' T = \begin{bmatrix} \sum_{i=1}^{m_2} t_{i1} t_{i2} & \sum_{i=1}^{m_2} t_{i1} t_{i2} & \dots & \sum_{i=1}^{m_2} t_{i1} t_{in_2} \\ \sum_{i=1}^{m_2} t_{i2} t_{i1} & \sum_{i=1}^{m_2} t_{i2} t_{i2} & \dots & \vdots \\ \vdots & \dots & \ddots & \vdots \\ \sum_{i=1}^{m_2} t_{in_2} t_{i1} & \dots & \dots & \sum_{i=1}^{m_2} t_{in_2} t_{in_2} \end{bmatrix} \quad (5)$$

Setiap *non-diagonal element* pada matriks H adalah Absolute Support ($SupA$) dari setiap anggota kandidat *frequent 2-itemset* (C_2) $SupA(C_2)$, sedangkan *diagonal element* adalah Absolute Support ($SupA$) setiap *item* pada *frequent 1-itemset* $Sup(L_1)$.

Setelah terbentuk matriks H , proses selanjutnya adalah kemudian membentuk *symetric matrix* $G = (g_{ij})$ yang dibentuk sesuai dengan Hermite matrix (H) untuk mendapatkan *frequent 2-itemset* (L_2), persamaan ini dapat dilihat pada persamaan 6.

$$G = g_{ij} = g_{ji} = \begin{cases} 1 & h_{ij} \geq Sup_{min} & i \neq j \\ 0 & h_{ij} < Sup_{min} & i \neq j \\ 0 & & i = j \end{cases} \quad (6)$$

Dimana:

- Nilai 1 diberikan pada elemen g_{ij} matriks G jika elemen h_{ij} pada matriks H kurang dari Sup_{min} dengan catatan, baris (i) tidak sama dengan kolom (j).

- Nilai 0 diberikan pada element g_{ij} jika elemen h_{ij} pada matriks H kurang dari Sup_{min} dengan catatan baris (i) tidak sama dengan kolom (j).

- Nilai 0 diberikan pada element g_{ij} matriks G jika i sama dengan j .

Frequent 2-itemset (L_2) didapatkan dari matriks G , dimana elemen g_{ij} bernilai 1, $L_2 = \{C_2 | g_{ij} = 1, i, j = 1, 2, \dots, P_1, i \neq j\}$. Langkah terakhir

baris dalam T yang jumlahnya kurang dari 3, $\sum_{j=1}^n t_{ij} < 3$ dibuang. Banyaknya baris dan kolom kolom dinotasikan dengan m_3 dan n_3 .

Definisi.6 Mining Frequent k-Itemsets

Pertama sebuah matriks U^k dibentuk untuk pembentukan kandidat k-Itemsets (C_k) yang berasal dari frequent itemset sebelumnya, frequent (k-1)-itemsets. Persamaan ini dapat dilihat pada persamaan 7.

$$U_k = (U_1^k, U_{2,...}^k, U_1^k)' = (C_k, C_{k,...}, C_k)' \quad (7)$$

Kemudian sebuah matriks W dibentuk dengan persamaan 8.

$$W = (w_{ij})_{m \times m} = U^k T_k' \quad (8)$$

Simbol:

W = matriks W

U^k = matriks U dengan k indeks

T = Transaction Matrix (T) dengan k indeks

Fungsi $F(W)$ dibentuk dari matriks W untuk memberikan nilai pada element matriks $W(w_{ij})$ dengan persamaan 9.

$$F(w_{ij}) = \begin{cases} 1, w_{ij} \geq k \\ 0, w_{ij} < k' \end{cases} \quad i, j = 1, 2, \dots, m \quad (9)$$

Dimana :

- Nilai 1 diberikan jika jumlah nilai $w_{ij} \geq k$ dimana w_{ij} merupakan Absolute Support ($SupA$) dari kandidat k-itemset yang dibentuk (C_k), k adalah indek dan nilainya sesuai dengan index k pada frequent k-itemset.
- Nilai 0 diberikan jika jumlah w_{ij} tidak sama dengan k.

Matrix fungsi $F(W)$ ditunjukkan pada persamaan dibawah 10.

$$F(W) = \begin{bmatrix} f\left[\sum_{i=1}^n u_{1i}t_{1i}\right] & \cdots & f\left[\sum_{i=1}^n u_{1i}t_{mi}\right] \\ \vdots & \ddots & \vdots \\ f\left[\sum_{i=1}^n u_{mi}t_{1i}\right] & \cdots & f\left[\sum_{i=1}^n u_{mi}t_{mi}\right] \end{bmatrix} \quad (10)$$

Setelah terbentuk matriks W kemudian dibentuk matriks vektor v untuk menyimpan nilai Sup dan $SupA$ kandidat k-itemset sesuai matrix W , ditunjukkan pada persamaan 11.

$$V = (v_i)_{m \times 1} = [\sum_{i=1}^m f(w_{1i}), \dots, \sum_{i=1}^m f(w_{mi})] \quad (11)$$

Dalam kolom vektor v , setiap elemen v_i ($v_i > 0$) sama dengan Absolute Support dari kandidat k-Itemset, $SupA(C_k)$ dan $Sup(C_k)$. Kandidat itemset yang kurang dari Support minimal (Sup_{min}) kemudian dibuang, maka didapatkan frequent k-itemset (L_k). Proses terakhir membuang elemen baris t_{ij} di T yang kurang dari $k+1$ ($\sum_{j=1}^n t_{ij} < k+1$).

Ketika $k=k+1$, jika jumlah anggota dari frequent(k-1)-itemset (L_{k-1}) kurang dari k , kandidat k-itemsets yang dihasilkan oleh L_{k-1} akan jadi kosong atau null sehingga proses dari frequent k-itemsets akan berhenti, jika tidak maka proses berlanjut ke pencarian itemset selanjutnya mining frequent (k+1)-itemset.

2.2 Interestingness Measure

Piatetsky-Shapiro mengajukan tiga prinsip Interestingness Rule^[6]:

1. Interestingness Rule adalah independence, jika hasilnya=0.
2. Interestingness Rule meningkat jika $\frac{n_a n_b}{n}$.
3. Interestingness Rule menurun ketika $-\frac{n_a n_b}{n}$.

Untuk membangkitkan association rule dari frequent itemset yang telah terbentuk, digunakan Interestingness Measure dari G. Piatetsky-Shapiro untuk mendapatkan nilai Interestingness dari kandidat rule ($R:A \rightarrow B$) yang dibentuk dari frequent k-itemset ($k \geq 2$) yang telah dibentuk dapat dilihat pada persamaan 11.

$$I = n_{ab} - \frac{n_a n_b}{n} \quad (11)$$

Dimana

I = Interestingness Rule

n_{ab} = jumlah kemunculan bersama Antecedent dan Consequent ($SupA$)

n_a = Absolute Support Antecedent, $SupA$ (Antecedent)

n_b = Absolute Support Consequent, $SupA$ (Consequent)

4. Prosedur MibARM

```

Input:
Database=D
Banyak Transaksi yang berbeda=m
Set of n distinct items=I
Minimum Support= $Sup_{min}$ 
Minimum Interestingness=  $I_{min}$ 
Set
candidate itemset  $C_k = \Phi$ 
Frequent itemset  $L_k = \Phi$ 
Association rules set  $R = \Phi$ 
Transform database D to matrix T
Begin
Remove columns that  $\sum_{i=1}^m tij < m \cdot supmin$ 
Let  $C_1=I$ ,  $L_1=\Phi$ 
For j=1; j ≤ n ; j++
    Calculate the Sup( $I_j$ )
    If Sup( $I_j$ ) <  $Sup_{min}$ 
        Delete  $I_j$  from  $C_1$ 
    Endif
Set  $L_1=C_1$ ,
number of rows:  $m_1$ 
number of columns:  $n_1$ .
Remove rows that  $\sum_{j=1}^n tij < 2$ 
Let  $C_2=\{I_i I_j | i, j=1, 2, \dots, m_1\}$ ,  $L_2=\Phi$ 
Do{
    Construct Hermite matrix H
    If Sup( $I_i I_j$ ) <  $Sup_{min}$ 
        Delete  $I_i I_j$  from  $C_2$ 
    Endif
}
While(Length of  $C_2$  Doesn't change )
Let  $L_2=C_2$ 
number of rows:  $m_2$ 
number of coloumns:  $n_2$ 
Remove rows that  $\sum_{j=1}^n tij < 3$ 
Let  $C_k=\{C_1 | l=1, 2, \dots, m_2\}$ ,  $L_k=\Phi$ 
do{
    do{
        Construct matrix W
        Calculate the Sup( $C_k$ )
        If Sup( $c_1$ ) <  $Sup_{min}$ 
            Delete  $C_1$  from  $C_k$ 
        endif
    }while (Length of  $C_k$  doesn't
change)
    Set  $L_k = C_k$ 
    number of rows:  $m_k$ 
    number of columns:  $n_k$ 
    Remove rows that  $\sum_{j=1}^n tij < k + 1$ 
}While ( $C_k=\Phi$ )
Generate rules  $R_c$  by  $L_2, L_3, \dots, L_k$ 
Do {
    Calculate the  $I(r_i)$ 
    If  $I(r_i) > I_{min}$ 
        Add  $r_i$  into R

```

```

endif
}while ( $R_c \neq \Phi$ )
Output R
end

```

5. Evaluasi MibARM

5.1 Skenario 500 Transaksi

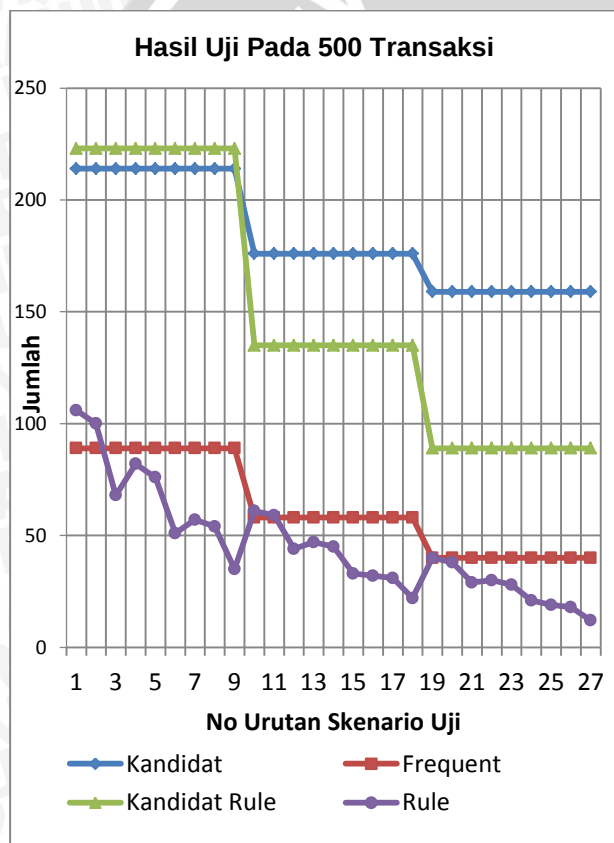
Pada skenario ini digunakan untuk menguji metode MibARM menggunakan data transaksi berjumlah 500 Transaksi pembelian, untuk hasil pengujian ini dapat dilihat padat tabel 1 dan untuk hasil yang lebih rinci dapat dilihat pada lampiran hasil uji coba pada 500 transaksi.

Tabel 1 Hasil Uji Coba Pada 500 Transaksi

	N	S	u	Sup	Conf	Im	Juml	Juml	Jumlah	Jum
	0	ji	mi	mi	mi	in	ah	ah	Kandidat	lah
			n	n	n		Kand	Freq	Rule	Rul
							idat	uent		e
							Items	Items		
1	1	1	0.2	0.5	214	89	223	106		
2	2	2	0.2	1	214	89	223	100		
3	3	3	0.2	5	214	89	223	68		
4	4	4	0.4	0.5	214	89	223	82		
5	5	5	0.4	1	214	89	223	76		
6	6	6	0.4	5	214	89	223	51		
7	7	7	0.6	0.5	214	89	223	57		
8	8	8	0.6	1	214	89	223	54		
9	9	9	0.6	5	214	89	223	35		
10	1	1	0.2	0.5	176	58	135	61		
11	1	2	0.2	1	176	58	135	59		
12	1	3	0.2	5	176	58	135	44		
13	1	4	0.4	0.5	176	58	135	47		
14	1	5	0.4	1	176	58	135	45		
15	1	6	0.4	5	176	58	135	33		
16	1	7	0.6	0.5	176	58	135	32		
17	1	8	0.6	1	176	58	135	31		
18	1	9	0.6	5	176	58	135	22		
19	1	1	0.2	0.5	159	40	89	40		
20	2	2	0.2	1	159	40	89	38		
21	2	3	0.4	5	159	40	89	29		
22	2	4	0.4	0.5	159	40	89	30		

No	S	uji	Support	Confidence	Minimum	Jumlah Kandidat Itemset	Jumlah Frequent Itemset	Jumlah Kandidat Rule	Jumlah Rule
23		5			1	159	40	89	28
24		6			5	159	40	89	21
25		7			0.5	159	40	89	19
26		8			1	159	40	89	18
27		9			5	159	40	89	12

Line Graph sesuai dengan tabel 1 dapat dilihat pada gambar 1, dimana sumbu x adalah no urutan skenario uji coba, dan sumbu y adalah skalar nilai dari 0 sampai 250.



Gambar 1 Line Graph Uji Coba Pada 500 Transaksi

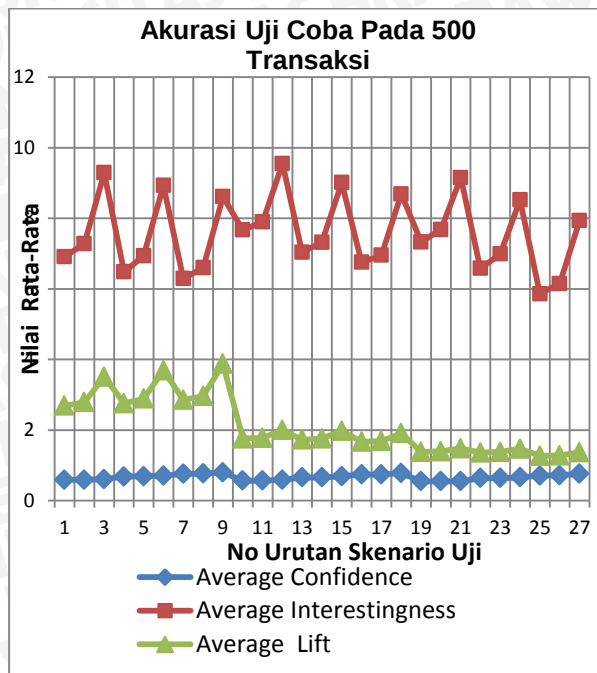
Seperti yang terlihat pada gambar 1, dapat diketahui bahwa semakin besar minimum Support yang diberikan maka semakin kecil kandidat frequent itemset yang terbentuk, sedangkan semakin besar nilai minimum Confidence dan minimum Interestingness maka semakin kecil jumlah

rule yang dihasilkan. Tabel Akurasi rule hasil Pembentukan ini dapat dilihat pada tabel 2.

Tabel 2 Akurasi Rule hasil Pembentukan

No	S	uji	Support	Confidence	Minimum	Average Confidence	Average Interestingness	Average Lift
1	1	1	0.02	0.2	0.5	0.59	6.91	2.69
2		1			0.59	7.28	2.79	
3		5			0.61	9.29	3.51	
4		4		0.4	0.5	0.68	6.48	2.76
5		1			0.69	6.94	2.89	
6		5			0.71	8.93	3.69	
7		7		0.6	0.5	0.76	6.29	2.86
8		1			0.77	6.6	2.96	
9		5			0.8	8.61	3.88	
10	2	1	0.03	0.2	0.5	0.57	7.67	1.76
11		1			0.57	7.9	1.78	
12		5			0.59	9.55	2	
13		4		0.4	0.5	0.66	7.04	1.73
14		1			0.66	7.32	1.76	
15		5			0.69	9.01	1.98	
16		7		0.6	0.5	0.74	6.75	1.67
17		1			0.74	6.95	1.69	
18		5			0.78	8.69	1.91	
19	3	1	0.04	0.2	0.5	0.55	7.33	1.38
20		1			0.55	7.68	1.4	
21		5			0.55	9.15	1.48	
22		4		0.4	0.5	0.64	6.58	1.36
23		1			0.64	6.99	1.38	
24		5			0.66	8.52	1.47	
25		7		0.6	0.5	0.71	5.86	1.27
26		1			0.72	6.15	1.28	
27		5			0.76	7.93	1.37	

Line Graph sesuai dengan tabel 2 dapat dilihat pada gambar 2, dimana sumbu x adalah no urutan skenario uji coba dan sumbu y adalah skalar nilai dari 0 sampai 12.



Gambar 2 Line Graph Akurasi Hasil Uji Pada 500 Transaksi

Sesuai dengan Gambar 2, dapat diketahui bahwa nilai rata-rata Interestingness pada *rule* akan semakin meningkat seiring meningkatnya nilai minimum Interestingness dan minimum Confidence yang diberikan sesuai dengan skenario pada tabel 2.

Untuk rata-rata akurasi *rule* dengan Lift Ratio diketahui bahwa semakin meningkat seiring dengan meningkatkan nilai Interestingness dan Confidence.

Nilai akurasi Lift Ratio tertinggi adalah pada saat skenario 1 uji coba 9, dengan nilai Support minimum 0.02, minimum Confidence 0.6 dan minimum Interestingness 5 dengan jumlah rule 35.

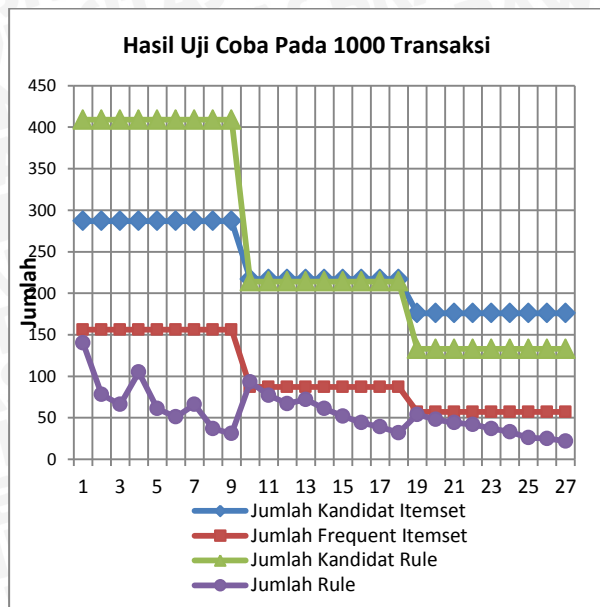
5.2 Skenario 1000 Transaksi

Pada skenario ini digunakan untuk menguji metode MIBARM menggunakan data transaksi berjumlah 1000 Transaksi pembelian, untuk hasil pengujian ini dapat dilihat pada tabel 3.

Tabel 3 Hasil Uji Coba Pada 1000 Transaksi

N o	S u ji	S u p m i n	C o n f m i n	I m i n	Juml ah Kand idat I tems et	Juml ah Freq uent I tems et	Jumlah Kandidat Rule	Jum lah Rul e
1	1	0.01	0.2	1	287	156	409	140
2	2			5	287	156	409	78
3	3			10	287	156	409	66
4	4		0.4	1	287	156	409	105
5	5			5	287	156	409	61
6	6			10	287	156	409	51
7	7		0.6	1	287	156	409	66
8	8			5	287	156	409	37
9	9			10	287	156	409	31
10	1	0.02	0.2	1	217	87	214	93
11	2			5	217	87	214	77
12	3			10	217	87	214	67
13	4		0.4	1	217	87	214	72
14	5			5	217	87	214	61
15	6			10	217	87	214	52
16	7		0.6	1	217	87	214	44
17	8			5	217	87	214	39
18	9			10	217	87	214	32
19	1	0.03	0.2	1	176	57	133	54
20	2			5	176	57	133	48
21	3			10	176	57	133	44
22	4		0.4	1	176	57	133	42
23	5			5	176	57	133	37
24	6			10	176	57	133	33
25	7		0.6	1	176	57	133	26
26	8			5	176	57	133	25
27	9			10	176	57	133	22

Line Graph sesuai dengan tabel 3 dapat dilihat pada gambar 3, dimana sumbu x adalah no urutan skenario uji coba, dan sumbu y adalah skalar nilai dari 0 sampai 450.



Gambar 3 Line Graph Hasil Uji Coba pada 1000 Transaksi

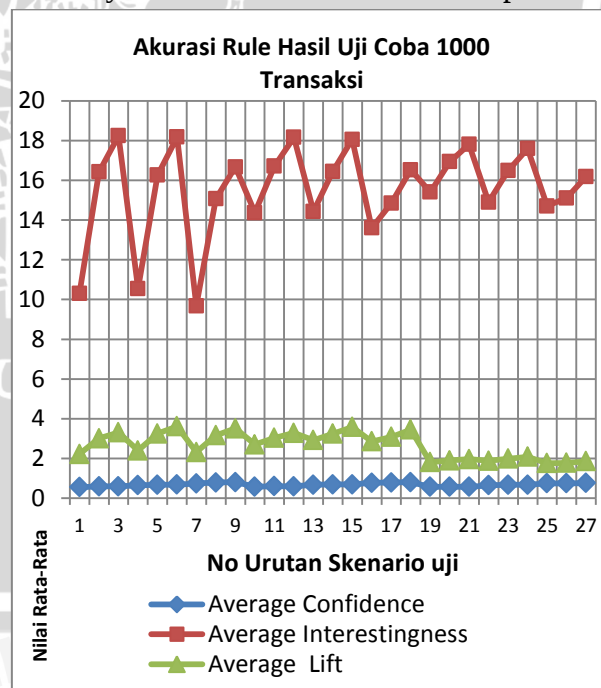
Seperti yang terlihat pada gambar 3, dapat diketahui bahwa semakin besar minimum Support yang diberikan maka semakin kecil kandidat *frequent itemset* yang terbentuk, sedangkan semakin besar nilai minimum Confidence dan minimum Interestingness maka semakin kecil jumlah *rule* yang dihasilkan. Tabel akurasi *rule* hasil Pembentukan pada 1000 transaksi dapat dilihat pada tabel 4.

Tabel 4 Tabel akurasi Rule Pada Uji 1000 Transaksi

N o	S u j i	Su p m i n	Co nf m i n	Imi n	Average Confide nce	Average Interesting ness	Avera ge Lift
1	1	0.01	0.2	1	0.55	10.31	2.22
2	2			5	0.59	16.42	3.01
3	3			10	0.59	18.24	3.31
4	4		0.4	1	0.65	10.55	2.4
5	5			5	0.67	16.26	3.25
6	6			10	0.68	18.18	3.61
7	7		0.6	1	0.74	9.68	2.32
8	8			5	0.79	15.07	3.16
9	9			10	0.8	16.67	3.51
10	1	0.02	0.2	1	0.58	14.37	2.7
11	2			5	0.6	16.71	3.03
12	3			10	0.59	18.16	3.28
13	4		0.4	1	0.67	14.43	2.92
14	5			5	0.69	16.44	3.24

N o	S u j i	Su p m i n	Co nf m i n	Imi n	Average Confide nce	Average Interesting ness	Avera ge Lift
15	6	0.6	0.6	10	0.69	18.06	3.58
16	7			1	0.77	13.6	2.86
17	8			5	0.79	14.84	3.08
18	9			10	0.8	16.53	3.45
19	1	0.2	0.2	1	0.57	15.41	1.81
20	2			5	0.58	16.94	1.9
21	3			10	0.57	17.81	1.96
22	4			1	0.65	14.9	1.88
23	5	0.03	0.4	5	0.67	16.49	1.99
24	6			10	0.67	17.6	2.08
25	7			1	0.75	14.71	1.76
26	8			5	0.75	15.11	1.78
27	9	0.6	0.6	10	0.76	16.19	1.86

Line Graph sesuai dengan tabel 4 dapat dilihat pada gambar 4, dimana sumbu x adalah no urutan skenario uji coba, dan sumbu y adalah skalar nilai dari 0 sampai 20.



Gambar 4 Line Graph Akurasi Hasil Uji Pada 1000 Transaksi

Dari hasil uji ini *rule* yang mempunyai rata-rata Lift Ratio tertinggi adalah pada saat diberikan kondisi sesuai dengan skenario 1 uji coba 6 dengan minimum Support 0.01,

minimum Confidence 0.6 dan minimum Interestingness 10 dengan jumlah Rule 51.

6. Analisis Hasil

Dari hasil uji coba yang telah dilakukan dilakukan sebelumnya kemudian dilakukan analisis lebih lanjut untuk mengetahui hasil yang didapatkan.

6.1 Analisis Pembentukan Frequent Itemset Menggunakan Metode MibARM

Setelah dilakukan implementasi dan ujicoba, dalam proses pembentukan *frequent itemset*, dapat diketahui bahwa algoritma MibARM dapat digunakan untuk menghindari terjadinya perulangan kandidat *frequent itemset* seperti terlihat pada contoh tabel 5.

Tabel 5 Contoh Perulangan Kandidat Itemset

Kandidat Frequent	Kandidat Perulangan
1,2	(2,1)
1,2,3	(1,3,2), (2,3,1), (3,1,2), (2,1,3), (3,2,1)
1,2,3,4	(1,2,4,3), (1,3,2,4), (1,3,4,2), (1,4,2,3), (1,4,3,2), (2,1,3,4), (2,1,4,3), (2,3,1,4), (2,3,4,1), (2,4,1,3), (2,4,3,1), (3,1,2,4), (3,1,4,2), (3,2,1,4), (3,2,4,1), (3,4,1,2), (3,4,2,1), (4,1,2,3), (4,1,3,2), (4,2,1,3), (4,2,3,1), (4,3,1,2)

Metode yang membantu dalam proses iterasi pembentukan kandidat itemset seperti penggunaan matriks T , jumlah transaksi dalam matriks T akan selalu berkurang karena ada pemangkasan baris transaksi disetiap iterasi pembentukan kandidat *itemset*, sehingga kandidat-kandidat yang tidak memenuhi kondisi dalam setiap iterasi akan dibuang.

Pada Iterasi pembentukan kandidat 2-*Itemset*, digunakan matriks Hermite (H) yang merupakan kombinasi dari *item* dalam transaksi, dimana setelah dibentuk kandidat

item, kemudian dicari jumlah kemunculan kandidat tersebut ($SupA$), matriks G digunakan untuk menyimpan nilai dari kandidat bila lebih besar dari minimum Support maka akan bernilai 1, jika kurang maka bernilai 0 dan bila kombinasi tersebut merupakan *item* yang sama maka juga bernilai 0, sehingga dalam pembentukan *frequent itemset* cukup mengambil kombinasi *item* dari matriks G yang bernilai satu sehingga metode ini juga membantu dalam mengurangi terjadinya perulangan kandidat *frequent itemset*.

Pada iterasi pembentukan kandidat *frequent* yang lebih dari 2-*itemset* metode MibARM membentuk kandidat yang berasal dari *frequent itemset* sebelumnya, sehingga tidak perlu mencari kandidat dari awal, dengan mengkombinasikan kandidat sebelumnya dengan *item* dalam Transaction matrix (T) yang telah dipangkas pada pembentukan *frequent itemset* sebelumnya, maka metode ini juga membantu mengurangi terjadinya perulangan kandidat *frequent itemset*.

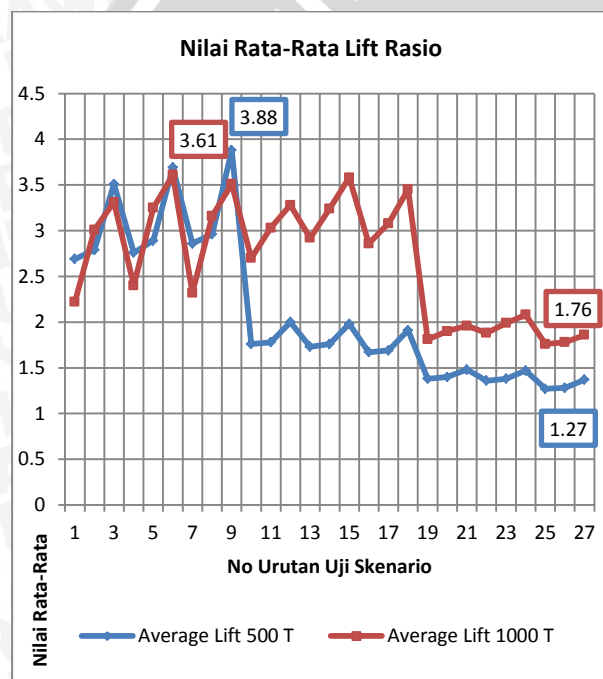
6.2 Analisis Pembentukan Rule

Hasil uji coba pada 500 dan 1000 data transaksi didapatkan *rule* yang memiliki rata-rata nilai Confidence 0.67 menunjukkan bahwa *rule* yang dihasilkan mempunyai peluang kemunculan lebih dari 50%. Rata-rata nilai Interestingness hasil uji coba pada 500 dan 1000 data transaksi 7.57 dan 15.54, ini menunjukkan bahwa *rule* yang dihasilkan mempunyai nilai Interestingness lebih besar dari 0.

Dari rata-rata nilai Lift Ratio hasil uji pada 500 transaksi dan 1000 transaksi, diketahui pada saat pembentukan *rule* dengan menggunakan Confidence dan Interestingness, nilai Lift Ratio dari setiap *rule* yang dihasilkan lebih besar dari 1 seperti yang terlihat pada gambar 5, ini menunjukkan bahwa kejadian pada setiap *rule* Antecedent mempunyai efek positif terhadap kejadian di setiap *rule* Consequent, sehingga semakin tinggi nilai Lift pada *rule* semakin tinggi

keterkaitan antara *rule* Antecedent dan *rule* Consequent. Semakin tinggi nilai Confidence yang dimiliki oleh *rule* maka semakin tinggi nilai Lift Ratio *rule* tersebut, demikian juga semakin tinggi nilai Confidence maka semakin tinggi nilai Interestingness *rule* tersebut. Semakin besar jumlah transaksi semakin tinggi juga nilai Interestingness Rule.

Rata-rata Lift Ratio tertinggi pada uji 500 transaksi adalah 3.88 pada saat nomor urutan uji skenario ke 9 dan terendah 1.37 pada nomor urutan uji skenario coba ke 25, dan rata-rata Lift Ratio tertinggi pada saat uji 1000 transaksi adalah 3.61 pada nomor urutan uji skenario ke 6 dan terendah 1.76 pada nomor urutan uji skenario ke 25. Rata-rata keseluruhan Lift Ratio pada hasil uji 500 dan 1000 Transaksi adalah 2.1 dan 2.67.



Gambar 5 Rata-Rata Lift Ratio Rule Hasil Uji

Referensi

- [1] Chan, Henry, Raymond Lee, Elizabeth Chang. 2001. *E-Commerce Fundamentals And Applications*. West Sussex: John Wiley & Sons, Inc.
- [2] Han, J.W, Micheline Kamber.2006. *Data Mining: Concept and Techniques*. New York: Morgan Kaufman Publisher.
- [3] Tsay, Yuh-Jiuan, Tain-Jung Hsu , Jing-Rung Yu . 2009. *FIUT: A New Method*

for Mining Frequent Itemset. ELSEVIER INC. www.elsevier.com

- [4] Song, Wei, Bingru Yang, Zhangyan Xu. 2008. *Index-BitTableFI: An Improved Algorithm for Mining Frequent Itemsets*. ELSEVIER INC. www.sciencedirect.com. Diakses Bulan Februari 2012.

- [5] Deng, Xiaoyi, Chun jin, Yoshiyuki Higuchi, Jim Han.2010. *An Effisient Association Rule Mining Method for Personalized Recommendation in Mobile E-Commerce*. New York: Atlantis Press. www.atlantispress.com. Diakses Bulan September 2011.

- [6] Lenca, Philippe, Patrick Meyer, Benoit Vaillant, Stephane Lallich.2008. *On selecting interestingness measures for association rules: User Oriented Description and Multiple Criteria Decision Aid*. ELSEVIER INC. www.sciencedirect.com. Diakses Bulan Mei 2012