

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

Dalam bab ini dibahas mengenai metodologi dan perancangan rekomendasi pembelian pada *web Commerce* menggunakan MIBARM. Berikut ini tahap-tahap yang dilakukan dalam penelitian ini:

1. Mempelajari metode yang akan digunakan dalam jurnal “*An Efficient Association Rule Mining Method for Personalized Recommendation in Mobile Commerce*” dan studi literatur yang terkait dengan rekomendasi menggunakan Association Rule Mining.
2. Mengumpulkan *dataset* yang dibutuhkan untuk penelitian, yaitu *dataset* penjualan di toko roti.
3. Merancang perangkat lunak untuk menerapkan algoritma MIBARM untuk rekomendasi pembelian pada *web Commerce*.
4. Implementasi perangkat lunak berdasarkan perancangan yang telah dilakukan.
5. Melakukan uji coba perangkat lunak yang telah dibuat menggunakan *dataset* transaksi penjualan.
6. Melakukan evaluasi terhadap perangkat lunak yang telah dibuat terhadap hasil *rule* yang telah dibentuk menggunakan Lift Ratio.

Untuk memberikan gambaran visual, langkah-langkah penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1 Langkah-langkah Penelitian

3.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* penjualan toko roti di USA yang menjual 40 jenis roti dan 10 jenis minuman, *dataset* ini dapat diunduh di alamat <https://wiki.csc.calpoly.edu/Datasets/wiki/ExtendedBakery>.

Setiap transaksi dalam *dataset* bisa memiliki beberapa *item* yang terjual, sehingga *dataset* ini bisa digunakan untuk pencarian Association Rule Mining.

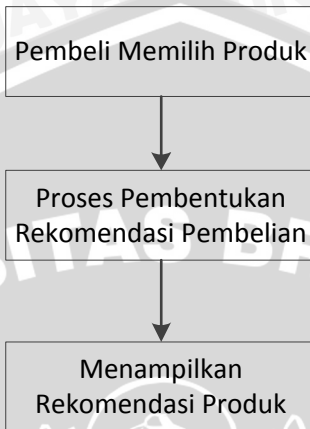
3.2 Deskripsi Sistem

Perangkat lunak yang akan dibuat menangani pencarian *association rule* dari *dataset* transaksi penjualan menggunakan algoritma MIBARM.

Rekomendasi yang dihasilkan dapat membantu pembeli ketika memilih suatu barang dengan menyarankan suatu barang yang mempunyai hubungan dengan barang tersebut. Berikut merupakan alur proses dalam sistem rekomendasi menggunakan algoritma MIBARM:

1. Pembeli memilih suatu produk yang akan dibeli yang ditampilkan pada *web Commerce*, ketika pembeli sudah memilih dan memasukkan produk dalam keranjang belanja kemudian sistem rekomendasi menggunakan MIBARM akan bekerja.
2. Sistem akan melakukan pencarian *frequent itemset*, kemudian sistem akan membentuk *rule* dari *frequent itemset* yang telah terbentuk berdasarkan Interestingness measure.
3. Kemudian akan ditampilkan rekomendasi pembelian berdasarkan produk yang telah dipilih oleh pembeli.

Rancangan dari alur sistem rekomendasi dapat dilihat pada gambar 3.2.



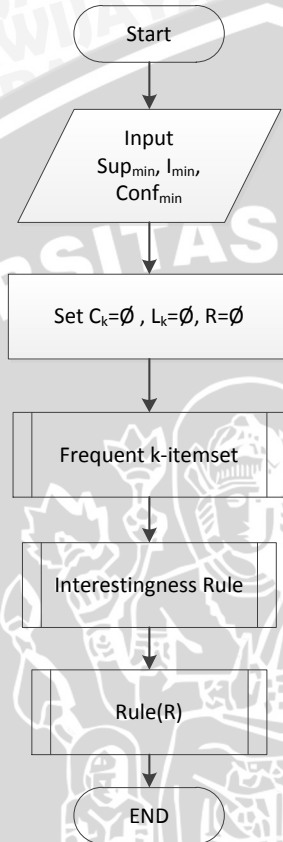
Gambar 3.2 Rancangan Alur Sistem Rekomendasi

3.3 Perancangan Sistem

Dalam proses menghasilkan rekomendasi menggunakan algoritma MibARM terdapat dua proses utama yaitu proses untuk menghasilkan *frequent itemset* dari *dataset* transaksi, proses yang kedua membentuk *rule* berdasarkan nilai Interestingness dari *frequent itemset* yang telah terbentuk. Berikut merupakan alur proses utama dari sistem rekomendasi menggunakan MibARM:

1. Diberikan nilai untuk Support minimal (Sup_{min}), Confidence minimal ($Conf_{min}$) dan Interestingness minimal (I_{min}).
2. Set kandidat *itemset* $C_k = \emptyset$, *frequent itemset* $L_k = \emptyset$ dan *association rule* $R = \emptyset$.
3. Proses pencarian kandidat *itemset* (C_k) untuk membentuk *frequent itemset* (L_k).
4. Membentuk *rule* berdasarkan nilai Confidence minimal ($Conf_{min}$) dan minimal *interestingness measure* (I_{min}) dari *frequent itemset* yang telah terbentuk.
5. Menampilkan output berupa *rule* untuk sistem rekomendasi sesuai produk yang dipilih pembeli.

Alur dari proses ini dapat dilihat pada gambar 3.3.



Gambar 3.3 Alur Proses untuk Pembentukan Proses Utama

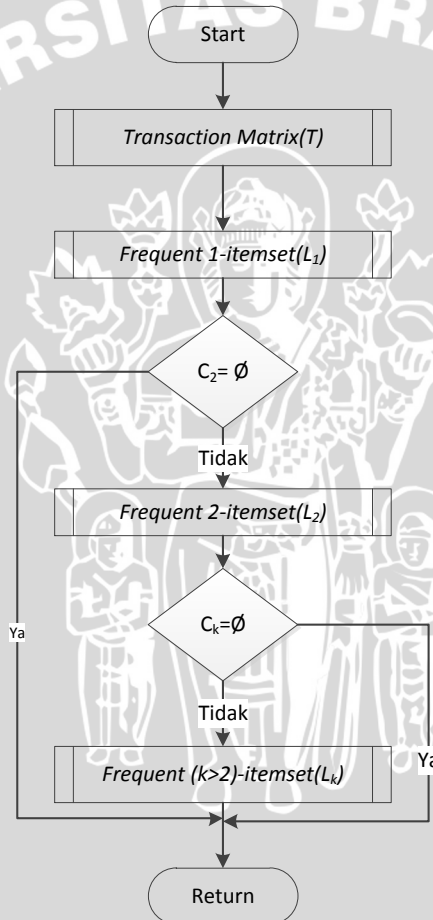
3.3.1 Pembentukan Frequent Itemset

Pada proses ini dilakukan sekali pemindaian *database* yang kemudian disimpan dalam sebuah Transaction Matrix (T). Proses selanjutnya adalah melakukan pencarian *frequent itemset*. Alur proses ini adalah:

1. Dataset transaksi penjualan dalam *database* diubah menjadi Transaction Matrix (T).
2. Pencarian kandidat *frequent 1-itemset* (C_1) dilakukan untuk mendapatkan *frequent 1-itemset* (L_1).

3. Setelah didapatkan L_1 kemudian dicari kandidat *frequent 2-itemset* untuk mendapatkan *frequent 2-itemset* (L_2).
4. Pada proses selanjutnya dilakukan pencarian kandidat yang lebih besar dari 2 *frequent* ($k > 2$)-*itemset*, proses ini akan berhenti jika kandidat *itemset* (C_k) kosong.

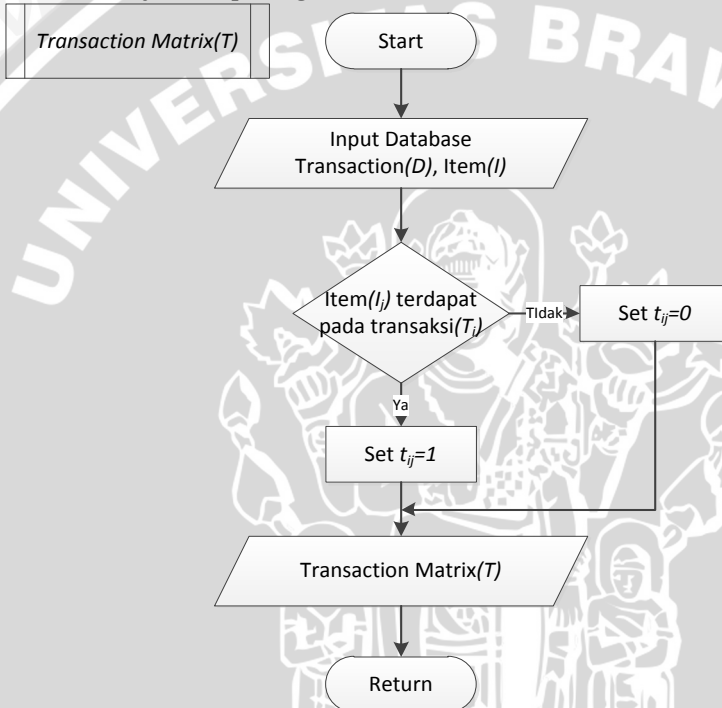
Alur proses ini dapat dilihat pada gambar 3.4.



Gambar 3.4 Alur Proses Pembentukan Frequent Itemset

3.3.1.1 Pembentukan Transaction Matrix (T)

Dataset transaksi dalam *database* diubah menjadi Transaction Matrix (*T*) menggunakan persamaan 2.1, dimana elemen t_{ij} pada matriks *T* akan bernilai 1 jika *item* (I_j) terdapat pada transaksi (T_i), jika tidak maka elemen t_{ij} akan bernilai 0. Alur proses pembentukan Matriks *T* ditunjukkan pada gambar 3.5.



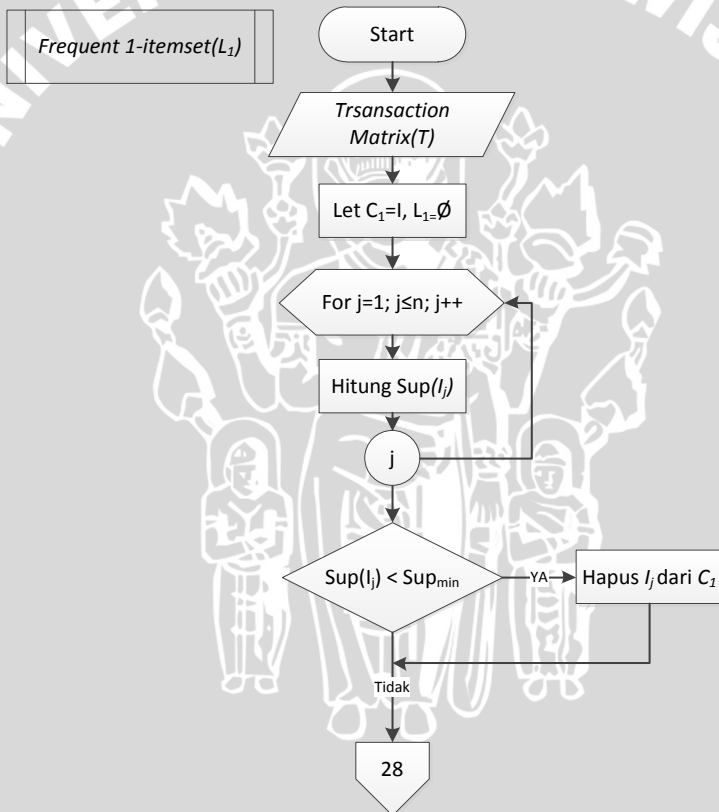
Gambar 3.5 Alur Proses Pembentukan Transaction Matrix (T)

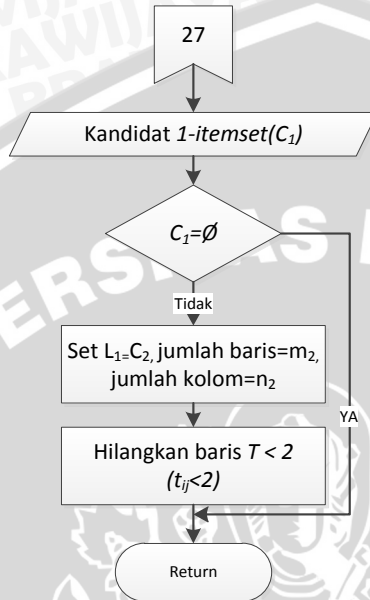
3.3.1.2 Pencarian Mining Frequent 1-Itemset

Transaction Matrix (*T*) yang telah dibentuk kemudian dilakukan proses pencarian kandidat *frequent 1-itemset* untuk membentuk *frequent 1-itemset* (L_1). Alur dari proses ini adalah:

1. Transaction Matrix (*T*) sebagai masukan data pada proses ini.
2. Kemudian di set kandidat 1-Itemset $C_1=I$, $L_1= \emptyset$.
3. Proses selanjutnya menghitung semua Support setiap *item* (i_j) pada kandidat C_1 sesuai dengan persamaan 2.3 dan 2.4, *item* (i_j)

- yang memiliki Support kurang dari Support minimal (Sup_{min}) dibuang sesuai dengan persamaan 2.5.
4. Jika didapatkan kandidat *1-Itemset* (C_1) maka di set $C_1 = L_1$ (*frequent 1-itemset*), jika C_1 kosong maka proses akan berhenti.
 5. Kemudian elemen baris t_{ij} pada T yang kurang dari 2 dihilangkan ($\sum_{j=1}^n t_{ij} < 2$).
- Alur proses pencarian *frequent Itemset* (L_1) dapat dilihat pada gambar 3.6.





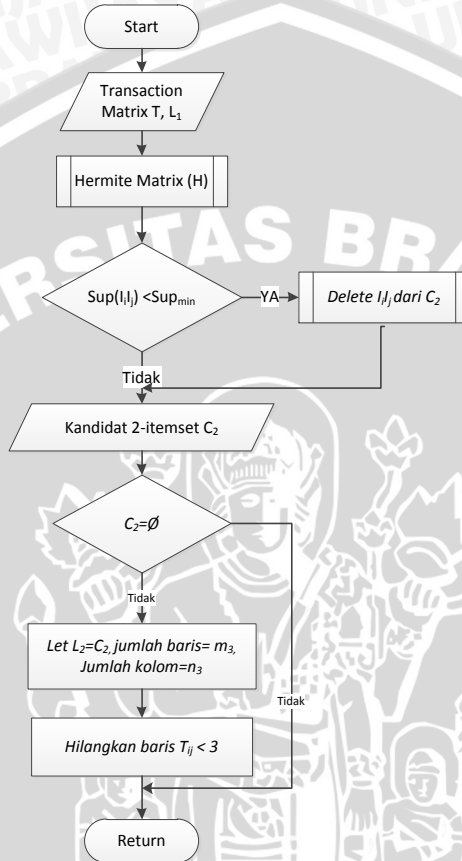
Gambar 3.6 Alur Proses Pencarian Frequent 1-Itemset

3.3.1.3 Pencarian Mining Frequent 2-Itemset

Proses selanjutnya adalah melakukan pencarian kandidat *frequent 2-itemset* (C_2). Alur dari proses ini adalah sebagai berikut:

1. Kandidat C_2 dibentuk dari pasangan 2 *item* dari Transaction Matrix (T), $C_2 = \{I_i I_j, j = 1, 2, \dots, m_1\}$.
2. Pasangan 2 *item* ini dibentuk dalam matriks Hermite (H) sesuai dengan persamaan 2.7.
3. Setelah terbentuk matriks Hermite (H) kemudian dilakukan penghitungan Support setiap kandidat 2-Itemset C_2 , jika tidak memenuhi dari minimal Support yang diberikan maka pasangan *item* ($I_i I_j$) akan dibuang dari kandidat *itemset*.
4. Jika kandidat *itemset* (C_2) kosong maka proses akan berhenti.
5. Kemudian C_2 dijadikan *frequent 2-itemset* (L_2).
6. Proses terakhir adalah menghilangkan elemen t_{ij} dalam T yang panjangnya kurang dari 3 ($t_{ij} < 3$).

Alur proses ini ditunjukkan pada gambar 3.7.

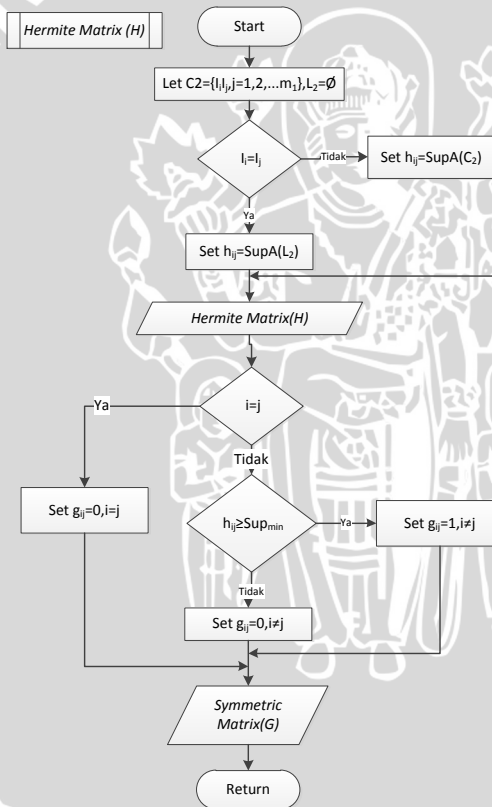


Gambar 3.7 Alur proses Pencarian Frequent 2-Itemset

Proses pembentukan matriks Hermite (H) adalah sebagai berikut:

1. Proses yang pertama adalah mendapatkan anggota kandidat 2-Itemset C_2 dari dari matriks T .
2. Jika baris $(i) =$ kolom (j) atau diagonal elemen maka elemen h_{ij} pada matriks H merupakan Absolute Support pada Frequent 1-Itemset $L_1, SupA(L_1)$.
3. Jika baris $(i) \neq$ kolom (j) tidak maka elemen h_{ij} adalah Absolute Support (SupA) setiap pasangan kandidat 2-Itemset $C_2, Sup(C_2)$.

4. Setelah didapatkan matriks Hermite (H) selanjutnya adalah membentuk matriks Symmetric (G) sesuai dengan matriks (H) dengan persamaan 2.7.
 5. Elemen g_{ij} pada matriks G akan bernilai 1 jika elemen h_{ij} pada matriks H memiliki Support yang lebih besar dari Sup_{min} dengan syarat $i \neq j$.
 6. Jika elemen h_{ij} memiliki Support yang lebih kecil dari Support minimal maka elemen g_{ij} akan bernilai 0 dengan syarat $i \neq j$.
 7. Jika $i = j$ maka elemen g_{ij} akan bernilai 0.
- Alur proses ini ditunjukkan pada gambar 3.8.



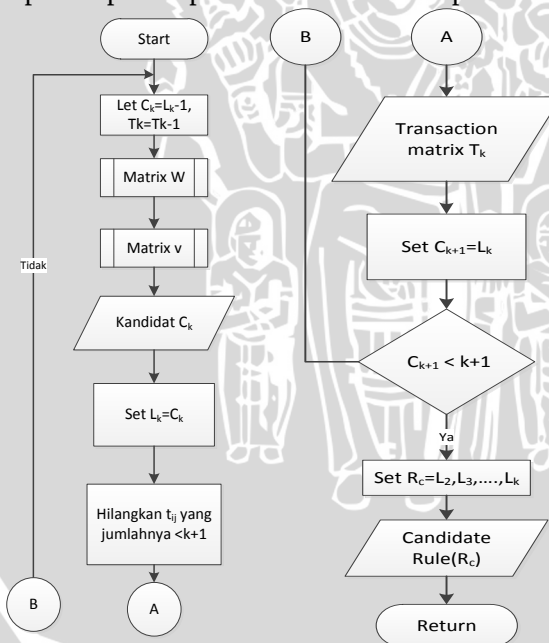
Gambar 3.8 Alur Proses Pembentukan matriks Hermite (H).

3.3.1.4 Pencarian Mining Frequent k-Itemset

Proses pencarian *frequent k-Itemset* ini akan terus berulang sampai kandidat C_k kurang dari $(k+1)$. Kandidat C_k dibentuk dari frequent *itemset* sebelumnya, k -Itemset. Alur dari proses ini adalah:

1. Didefinisikan kandidat k -Itemset $(C_k) = \{C_l | l=1,2,...,m2\}$ dan frequent k -itemset $L_k = \emptyset$.
2. Matriks W dibentuk sesuai persamaan 2.10, kemudian dihitung Support setiap kandidat *itemset* didalam W , kandidat yang tidak memenuhi Support minimal kemudian dibuang.
3. Jika panjang dari C_k sudah tidak berubah maka set $C_k = L_k$.
4. Proses selanjutnya adalah menghilangkan elemen t_{ij} dalam matriks T yang mempunyai jumlah kurang dari $k+1$ ($t_{ij} < k+1$).
5. Ketika kandidat jumlah C_k tidak memenuhi $(k+1)$ maka dibentuk *rule* (R_C) dari frequent *itemset* yang telah terbentuk ($L_2, L_3, ..., L_k$).
6. Output dari proses ini adalah kandidat *rule* (R_C).

Alur proses pencarian k -itemset dapat dilihat pada gambar 3.9.

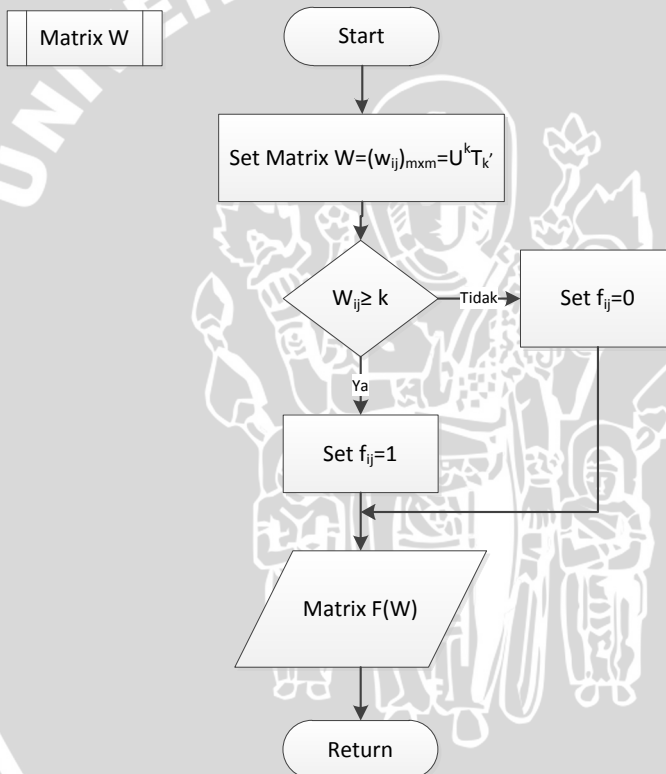


Gambar 3.9 Alur proses pembentukan Frequent k-Itemset

Alur proses pembentukan matriks W adalah sebagai berikut:

1. Langkah pertama adalah membentuk matriks (W) dengan persamaan 2.10.
2. Setiap elemen w_{ij} merupakan $SupA(C_k)$, kemudian fungsi $F(w_{ij})$ sesuai persamaan 2.11 digunakan untuk memberikan nilai pada w_{ij} , jika $w_{ij}=k$ maka $f(w_{ij})=1$, jika $w_{ij}<k$ maka $f(w_{ij})=0$.

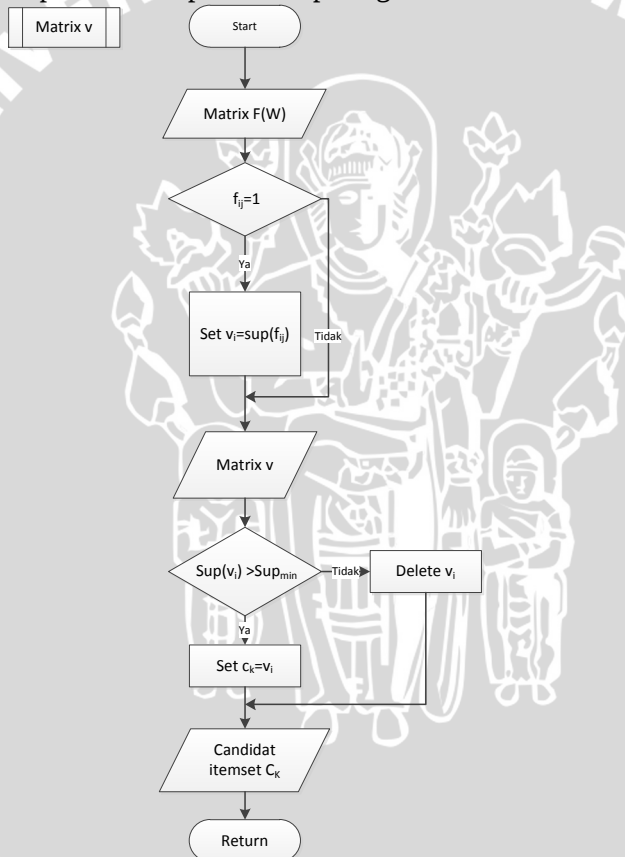
Alur proses ini ditunjukkan pada gambar 3.10.



Gambar 3.10 Alur proses Pembentukan Matrix W

Proses pembentukan matriks v adalah:

1. Matriks $F(w)$ merupakan inputan proses ini sesuai persamaan 2.13.
 2. Jika $f_{ij} \geq 1$ maka akan di set matrik $v_{ij} = \text{Sup}(f_{ij})$.
 3. Setelah didapatkan matriks v , langkah selanjutnya adalah menghilangkan v_i yang memiliki *Support* lebih kecil dari *Support* minimal yang ditetapkan.
 4. Langkah terakhir adalah set $C_k = v$.
- Alur proses ini dapat dilihat pada gambar 3.11.



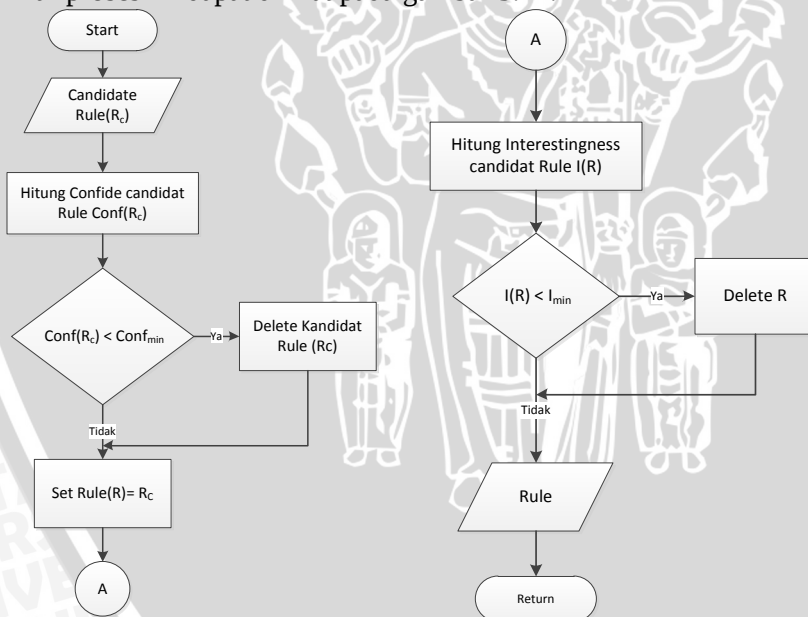
Gambar 3.11 Alur proses Pembentukan Matrix v

3.3.2. Pembentukan Rule berdasarkan Interestingness

Setelah kandidat *rule* terbentuk kemudian dilakukan proses perhitungan nilai Interestingness Rule untuk mengetahui keabsahan *rule* yang telah terbentuk, alur dari proses ini adalah:

1. Masukan proses adalah kandidat *rule* yang telah terbentuk ($RC_1, RC_2, \dots, RC_K$).
2. Setelah dihitung semua Confidence setiap kandidat *rule* maka kemudian kandidat *rule* yang kurang dari minimum Confidence dihilangkan.
3. Proses selanjutnya menghitung Interestingness Rule $I(R)$ berdasarkan persamaan 2.13 dan 2.14.
4. Jika Interestingness Rule kurang dari nilai minimal Interestingness yang telah ditetapkan maka *rule* dihilangkan.
5. Ketika anggota *rule* tidak kosong maka *output* adalah *rule*, jika kosong maka proses akan berhenti.

Alur proses ini dapat dilihat pada gambar 3.12.



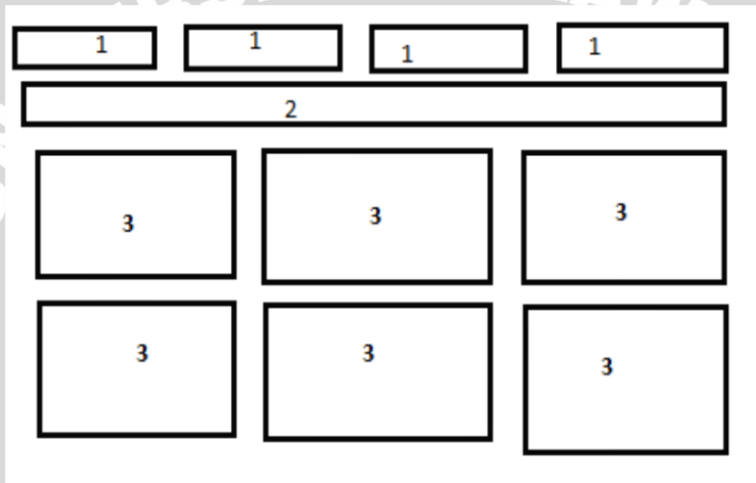
Gambar 3.12 Alur Proses Pembentukan Rule

3.4 Perancangan Desain Interface

Pada bagian ini akan dijelaskan tentang desain rancangan halaman *web* yang akan digunakan untuk mengimplementasikan perancangan sistem yang telah dibuat.

3.4.1 Perancangan Halaman Utama E-Commerce

Halaman utama *web* ini menampilkan kategori *item* yang dijual, gambar perancangan halaman ini dapat dilihat pada gambar 3.13.



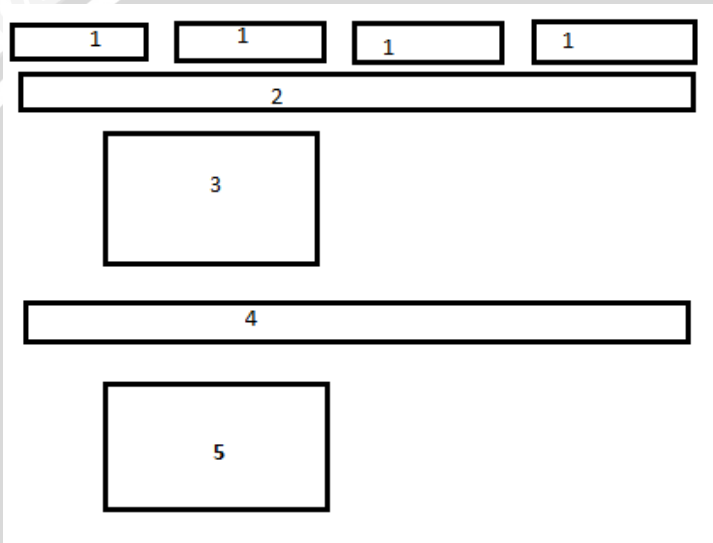
Gambar 3.13 Perancangan Halaman Utama

Keterangan gambar 3.13:

1. Menu:
 - Menu Halaman Utama
 - Menu Halaman Item Drink
 - Menu Halaman Item Food
 - Menu Halaman Pengaturan Rekomendasi
 - Menu Halaman Keranjang Belanja
2. Header : Berisi ucapan selamat datang
3. Content: Berisi daftar item-item yang dijual

3.4.2 Perancangan Halaman Rekomendasi Produk

Halaman ini menampilkan rekomendasi dari produk yang dijual, dan setelah setuju pembeli bisa memasukkan jumlah produk yang dibeli, ditampilkan juga rekomendasi *item* yg ada. Perancangan ini bisa dilihat pada gambar 3.14.



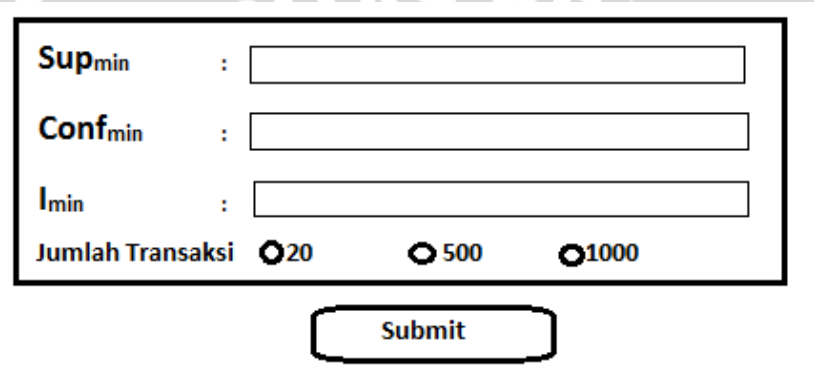
Gambar 3.14 Perancangan Halaman Rekomendasi Produk

Keterangan gambar 3.14:

1. Menu
2. Header
3. Content item yang dipilih Pembeli
4. Tag rekomendasi pembelian
5. Content Item yang direkomendasikan

3.4.3 Perancangan Halaman Pengaturan Rekomendasi

Halaman ini nantinya akan digunakan untuk mengatur parameter rekomendasi pembelian seperti minimum Support (Sup_{min}), minimum Confidence ($Conf_{min}$), minimum Interestingness (I_{min}) dan jumlah transaksi yang akan digunakan. Desain halaman ini bisa di gambar 3.15.



Sup_{min} :

Conf_{min} :

I_{min} :

Jumlah Transaksi ☒ 20 ☐ 500 ☐ 1000

Gambar 3.15 Perancangan Halaman Pengaturan Rekomendasi

3.5 Contoh Perhitungan Manual

Contoh data diambil dari *dataset* transaksi penjualan toko roti dengan mengambil 50 transaksi penjualan dan 50 *item* yang berbeda.

3.5.1 Transaction Matrik (T)

Dataset transaksi penjualan yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3.1 Dataset Transaksi

Receipt	Item	Flavor	Food	Type
1	7	Coffee	Eclair	Food
1	15	Blackberry	Tart	Food
1	49	Single	Espresso	Drink
1	44	Bottled	Water	Drink
2	1	Lemon	Cake	Food
2	19	Lemon	Tart	Food

Receipt	Item	Flavor	Food	Type
:	:	:	:	:
:	:	:	:	:
50	28	Tuile	Cookie	Food
50	39	Blueberry	Danish	Food

Transaksi pada tabel 3.4 kemudian disimpan dalam Transaction Matrix (T) yang dibentuk sesuai dengan persamaan 2.1, jika baris ke- i pada transaksi(T_i) terdapat kemunculan *item* (I_j) maka elemen t_{ij} bernilai 1, jika tidak maka bernilai 0, *item* yang digunakan adalah kategori *food* . Matrik ini dapat dilihat pada table 3.2,

Tabel 3.2 Screenshoot Tabel Transaction Matrix(T)

TID	[0]Cake	[1]Eclair	[2]Tart	...	[17] Espresso
1	0	1	1	...	1
2	0	1	1	...	0
:	:	:	:	:	:
20	0	1	1	...	0

3.5.2 Frequent 1-Itemset

Transaction Matrix (T) kemudian dilakukan penghitungan jumlah kemunculan setiap *item* dalam transaksi atau *supA* sesuai persamaan 2.3, hasil penghitungan ini dapat dilihat pada tabel 3.3.

$$-SupA(0)=t_{11}+t_{21}+...+t_{201}=0+0+...+0=0$$

$$-SupA(1)=t_{12}+t_{22}+...+t_{202}=1+1+...+1=15$$

$$-SupA(2)=t_{117}+t_{217}+...+t_{2017}=1+0+...+0=2$$

Tabel 3.3 Hasil Perhitungan SupA

TID	0	1	...	17
50	0	1	...	0
SupA	0	15	...	2

Setelah didapatkan Absolute Support (*SupA*) dari setiap *item*, proses selanjutnya adalah menghitung Support (*Sup*) setiap *item*

sesuai dengan persamaan 2.4, hasil perhitungan ini dapat dilihat pada tabel 3.4.

$$\text{-Sup}(0) = \frac{0}{20} = 0$$

$$\text{-Sup}(1) = \frac{6}{20} = 0.7$$

$$\text{-Sup}(17) = \frac{2}{20} = 0.1$$

Tabel 3.4 Hasil Perhitungan support (Sup)

TID	0	1	...	17
SupA	0	6	...	2
Sup	0	0.7	...	0.1

Untuk mendapatkan *frequent 1-itemset* L_1 maka digunakan persamaan 2.5 dimana setiap *item* yang memiliki nilai Support yang ditetapkan maka *item* tersebut dihilangkan. Minimum Support yang ditetapkan adalah 0.1, sehingga setiap *item* yang memiliki Support dibawah 0.1 dihilangkan. Table 3.5 menunjukkan *frequent 1-Itemset* yang didapat setelah proses penghilangan *item* yang memiliki Support dibawah minimal Support (sup_{min}) yang ditetapkan.

Tabel 3.5 Frequent 1-Itemset

L1	1	2	4	6	7	12	13	14	16	17
SupA	14	14	8	3	5	2	4	2	2	2
Sup	0.7	0.7	0.4	0.15	0.25	0.1	0.2	0.1	0.1	0.1

Setelah didapatkan *frequent 1-itemset* (L_1) proses selanjutnya adalah menghilangkan baris elemen transaksi pada matriks T yang jumlahnya kurang dari 2 ($\sum_{j=1}^n t_{ij} < 2$), hasil perhitungan ini dapat dilihat pada tabel 3.6.

$$\text{- Jumlah } T_1 = t_{11} + t_{12} + \dots + t_{117} = 4$$

$$\text{- Jumlah } T_2 = t_{22} + t_{22} + \dots + t_{217} = 2$$

$$\text{- Jumlah } T_{19} = t_{191} + t_{192} + \dots + t_{1917} = 4$$

Tabel 3.6 Hasil Perhitungan Panjang Baris Elemen t_{ij}

1	1	2	...	17	$\sum_{j=1}^n$
1	1	1	...	1	4
2	1	1	...	0	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
20	1	1	...	0	4

Setelah dihilangkan baris elemen t_{ij} yang jumlahnya kurang dari 2, maka didapatkan Transaction Matrix T .

3.5.3 Frequent 2-Itemset

Proses pertama dalam pencarian kandidat 2-Itemset (C_2) adalah membentuk Hermite matrix (H) sesuai dengan persamaan 2.6, dimana matrik ini dibentuk dari pasangan 2 itemset pada matriks T pada tabel 3.7.

Tabel 3.7 Transaction Matrik (T)

TID	1	2	...	17
1	1	1	...	1
2	1	1	...	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
20	1	1	...	0

Contoh perhitungan untuk mendapatkan nilai setiap elemen dari matriks H adalah:

$h_{11} = \text{SupA}(1) = 3$, karena merupakan diagonal elemen dan merupakan item yang sama $i=j$.

$h_{12} = (t_{11} * t'_{12}) + (t_{12} * t'_{22}) + (\dots) + (t_{119} * t'_{192}) = (0*0) + (0*1) + (\dots) + (0*0) = 10$

$h_{110} = (t_{11} * t'_{119}) + (t_{12} * t'_{219}) + (\dots) + (t_{119} * t'_{1019}) = (0*0) + (0*0) + (\dots) + (0*0) = 1$.

$h_{21} = (t_{21} * t'_{11}) + (t_{22} * t'_{12}) + (\dots) + (t_{219} * t'_{119}) = (0*0) + (0*1) + (\dots) + (0*0) = 10$

Hermite matrix (H) dapat dilihat pada tabel 3.8

Tabel 3.8 Hermite Matrix (H)

	1	2	...	17
1	14	10	...	1
2	10	14	...	1
⋮	⋮	⋮	⋱	⋮
17	1	1	...	1

Untuk mendapatkan *frequent 2-itemset* maka proses selanjutnya adalah membentuk matriks *Symmetric (G)* sesuai dengan *Hermite Matrix (H)* dengan persamaan 2.7. Contoh perhitungan untuk memberikan nilai pada elemen matriks (*G*) adalah:

- $g_{11}=0$, baris (i) sama dengan kolom (j).

$$-g_{12} = \frac{h_{12}}{m} = \frac{0}{20} = 0.$$

$$-g_{110} = \frac{h_{110}}{m} = \frac{1}{20} = 0.05$$

$$-g_{21} = \frac{h_{21}}{m} = \frac{10}{20} = 0.5$$

- $g_{22}=0$, baris (i) sama dengan kolom (j).

$$-g_{210} = \frac{h_{210}}{m} = \frac{1}{20} = 0.05$$

Symmetric matrix (G) dapat dilihat pada tabel 3.9.

Tabel 3.9 Symmetric Matrix (G)

	1	2	...	17
1	0	1	...	0
2	1	6	...	0
⋮	⋮	⋮	⋱	⋮
17	0	0	...	0

Dari *Symmetric matrix (G)* didapatkan *frequent 2-itemset* diambil dari matriks *G* yang bernilai 1

$L_2 = \{C_2 | g_{ij}=1, i,j=1,2,...,P_1, i \neq j\}$. Hasil proses pencarian *frequent 2-Itemset* L_2 dapat dilihat pada tabel 3.13.

Tabel 3.10 Frequent 2-Itemset L2

L2	SupA	Sup
1,7	5	0.25
1,13	3	0.15
2,1	10	0.5
2,7	4	0.2
2,13	4	0.2
2,16	2	0.1
4,1	5	0.25
4,2	4	0.2
6,1	2	0.1
6,2	3	0.15
6,7	2	0.1

Setelah didapatkan *frequent 2-itemset* (L_2) proses selanjutnya adalah menghilangkan baris elemen transaksi pada matriks T yang jumlahnya kurang dari 3 (**3**), hasil perhitungan ini dapat dilihat pada tabel 3.11.

- Jumlah $T_1 = t_{11} + t_{12} + \dots + t_{110} = 4$

- Jumlah $T_4 = t_{22} + t_{23} + \dots + t_{210} = 2$

- Jumlah $T_{19} = t_{191} + t_{192} + \dots + t_{1910} = 4$

Tabel 3.11 Hasil Perhitungan Panjang Baris Elemen t_{ij}

TID	1	2	...	17	$\sum_{j=1}^n$
0	1	1	...	1	4
1	1	1	...	0	4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	1	1	...	0	4

Setelah dihilangkan baris elemen t_{ij} yang jumlahnya kurang dari 3, maka didapatkan Transaction Matrix T dengan jumlah baris (m_3) adalah 11 dan jumlah kolom (n_3) adalah 10.

3.5.4 Frequent k-Itemset (L_k) (L_3)

Proses selanjutnya adalah pencarian kandidat *frequent 3-itemset* C_3 . Langkah pertama adalah membentuk kandidat *3-itemset* C_3 sesuai persamaan 2.9. Matriks U^3 dari *frequent Itemset* iterasi sebelumnya atau *k-itemset*, matriks ini dapat dilihat pada tabel 3.12.

Tabel 3.12 Matriks U_3

uk'	1,7	1,13	...	6,7
0	0	1	...	0
1	1	1	...	0
2	0	0	...	0
.	...		...	
10	1	0		1

Setelah terbentuk matriks (U^3) kemudian dibentuk matriks (W) sesuai dengan persamaan 2.10, sebelum dikalikan terlebih dahulu matriks (U^3) dilakukan *Transpose* yang dapat dilihat pada tabel 3.16 dan *Transaction Matrix* (T_3) juga di *Transpose* yang dapat dilihat pada tabel 3.13.

Tabel 3.13 Matrik U_3 Transpose

uk'	0	1	...	10
1,7	0	1	...	1
1,13	1	1	...	0
2,1	1	1	...	1
⋮	⋮	⋮	...	
6,7	0	0	...	1

Tabel 3.14 Transaction Matrix (T3) Transpose

T'	0	1	2	3	4	5	6	7	8	9	10
1	1	1	0	1	1	0	1	0	1	1	1
2	1	1	1	1	1	1	0	1	1	1	1
4	0	0	1	0	0	0	1	0	1	1	0
6	0	0	0	1	0	0	0	1	0	0	1
7	0	1	0	1	1	0	1	0	0	0	1
12	0	0	0	0	0	1	0	0	0	0	0
13	1	1	1	1	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	1	0	1	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0

Contoh perhitungan matriks (W) adalah:

$$-w_{11}=(u_{11}*t_{11})+(u_{12}*t_{12})+\dots+(u_{111}*t_{11})=(0*0)+(1*0)+\dots+(0*1)=5$$

$$-w_{12}=(u_{11}*t_{21})+(u_{12}*t_{22})+\dots+(u_{111}*t_{211})=(0*0)+(1*0)+\dots+(0*0)=4$$

$$-w_{111}=(u_{11}*t_{111})+(u_{12}*t_{112})+\dots+(u_{135}*t_{3535})=(0*0)+(1*0)+\dots+(0*0)=0$$

Matriks (W) dapat dilihat pada tabel 3.15.

Tabel 3.15 Matriks (W)

w	1	2	4	6	7	12	13	14	16	17
1,7	5	4	1	2	5	0	2	0	1	0
1,13	3	3	0	1	2	0	3	0	1	1
2,1	7	7	2	2	4	0	3	0	1	1
2,7	4	4	0	2	4	0	2	0	1	0
2,13	3	4	1	1	2	0	4	0	1	1
2,16	1	2	0	1	1	1	1	0	2	0
4,1	3	2	3	0	1	0	0	0	0	0

w	1	2	4	6	7	12	13	14	16	17
4,2	2	3	3	0	0	0	1	0	0	0
6,1	2	2	0	2	2	0	1	0	1	0
6,2	2	3	0	3	2	0	1	1	1	0
6,7	2	2	0	2	2	0	1	0	1	0

Setelah matriks (W) terbentuk proses selanjutnya adalah memberikan fungsi (F) pada matriks (W) sesuai dengan persamaan 2.11, contoh perhitungan $F(W)$ adalah:

- $f(w_{11}) = w_{11} = 5 = 5 > 3$ maka $f(w_{11}) = 1$

- $f(w_{12}) = w_{12} = 4 = 4 > 3$ maka $f(w_{12}) = 1$

- $f(w_{14}) = w_{14} = 3 = 3 = 3$ maka $f(w_{14}) = 1$

- $f(w_{10}) = w_{10} = 3 = 0 < 3$ maka $f(w_{10}) = 0$

Matrik $F(W)$ dapat dilihat pada tabel 3.16, dengan catatan jika ada perulang kandidat yang sama juga bernilai 0.

Tabel 3.16 $F(W)$

Fw	1	2	4	6	7	12	13	14	16	17
1,7	0	4	0	0	0	0	0	0	0	0
1,13	0	3	0	0	0	0	0	0	0	0
2,1	0	0	0	0	4	0	3	0	0	0
2,7	4	0	0	0	0	0	0	0	0	0
2,13	3	0	0	0	0	0	0	0	0	0
2,16	0	0	0	0	0	0	0	0	0	0
4,1	0	0	0	0	0	0	0	0	0	0

4,2	0	0	0	0	0	0	0	0	0	0
6,1	0	0	0	0	0	0	0	0	0	0
6,2	0	0	0	0	0	0	0	0	0	0
6,7	0	0	0	0	0	0	0	0	0	0

Kemudian matriks kolom *vector* (v) dibentuk untuk mendapatkan *frequent 3-itemset* (L_3) sesuai dengan persamaan 2.12, dengan terlebih dahulu dilakukan pengecekan terhadap duplikasi kandidat. Matriks ini dapat dilihat pada tabel 3.17.

Tabel 3.17 Matriks Kolom Vector (v)

v	SupA	Sup
1,7,2	4	0.2
1,13,2	3	0.15

Pada matriks (v) setiap kolom ($v_i > 0$) maka nilainya adalah Absolute Support dari kandidat *3-Itemset*, kemudian elemen v_i yang kurang dari minimum Support (Sup_{min}) dihilangkan sehingga didapatkan *frequent 3-itemset* (L_3). Hasil pembentukan *frequent 3-itemset* dapat dilihat pada tabel 3.18

Tabel 3.18 Frequent 3-Itemset (L_3).

L_3	SupA	Sup
1,7,2	4	0.2
1,13,2	3	0.15

Setelah didapatkan *frequent 3-itemset* (L_3) proses selanjutnya adalah menghilangkan baris elemen transaksi pada matriks T yang jumlahnya kurang dari $k+1$ ($3+1=4$), hasil perhitungan ini dapat dilihat pada tabel 3.19.

- Jumlah $T_1 = t_{11} + t_{12} + \dots + t_{110} = 4$
- Jumlah $T_2 = t_{22} + t_{22} + \dots + t_{210} = 4$
- Jumlah $T_{11} = t_{111} + t_{112} + \dots + t_{1110} = 4$

Tabel 3.19 Hasil Perhitungan Panjang Baris Elemen t_{ij}

T	1	2	4	6	7	12	13	14	16	17	
0	1	1	0	0	0	0	1	0	0	1	4
1	1	1	0	0	1	0	1	0	0	0	4
2	0	1	1	0	0	0	1	0	0	0	3
3	1	1	0	1	1	0	1	0	1	0	6
4	1	1	0	0	1	0	0	0	0	0	3
5	0	1	0	0	0	1	0	0	1	0	3
6	1	0	1	0	1	0	0	0	0	0	3
7	0	1	0	1	0	0	0	1	0	0	3
8	1	1	1	0	0	0	0	0	0	0	3
9	1	1	1	0	0	0	0	0	0	0	3
10	1	1	0	1	1	0	0	0	0	0	4

Setelah dihilangkan baris elemen t_{ij} yang jumlahnya kurang dari 4, karena kandidat *rule* hanya 2 maka tidak memenuhi $k+1$ ($3+1$)=4, maka proses iterasi berhenti.

3.5.5 Interestingness Rule

Frequent itemset yang telah didapatkan kemudian dilakukan perhitungan Confidence sesuai dengan persamaan 2.16, hasil perhitungan Confidence ini ditunjukkan pada tabel 3.20.

Contoh perhitungan Confidence:

$$\text{Conf } 1 \Rightarrow 7 = \frac{\text{SupA}(A,B)}{\text{SupA}(A)} = \frac{5}{14} = 0.36$$

Tabel 3.20 Confidence Rule

Rc	SupA	Sup	Conf
1=>7	5	0.25	0.36
7=>1	5	0.25	1
2=>1	10	0.5	0.71
1=>2	10	0.5	0.71
2=>7	4	0.2	0.29
7=>2	4	0.2	0.8
1=>13	3	0.15	0.21

Rc	SupA	Sup	Conf
13=>1	3	0.15	0.75
2=>13	4	0.2	0.29
13=>2	4	0.2	1
2=>16	2	0.1	0.14
16=>2	2	0.1	1
4=>1	5	0.25	0.63
1=>4	5	0.25	0.36
4=>2	4	0.2	0.5
2=>4	4	0.2	0.29
6=>1	2	0.1	0.67
1=>6	2	0.1	0.14
6=>2	3	0.15	1
2=>6	3	0.15	0.21
6=>7	2	0.1	0.67
7=>6	2	0.1	0.4
1,7=>2	4	0.2	0.8
1,2=>7	4	0.2	0.4
2,7=>1	4	0.2	1
1,13=>2	3	0.15	1
1,2=>13	3	0.15	0.3
2,13=>1	3	0.15	0.75

Setelah kandidat *rule* yang mempunyai nilai Confidence kurang dari minimum Confidence yang ditetapkan yaitu 0.2, maka didapatkan *rule*. Proses selanjutnya adalah menghitung nilai Interestingness Rule sesuai dengan persamaan 2.14. Hasil perhitungan Interestingness Rule ini dapat dilihat pada tabel 3.21. Contoh Perhitungan:

$$I(1=>7) = n_{ab} - \frac{n_a n_b}{n} = \text{SupA}(1 \Rightarrow 7) - \frac{\text{SupA}(1) * \text{SupA}(7)}{20} = 5 - \frac{14 * 5}{20} = 1.5$$

Tabel 3.21 Interestingness Rule

<i>Rule</i>	Conf	IR
$1 \Rightarrow 7$	0.36	1.5
$7 \Rightarrow 1$	1	1.5
$2 \Rightarrow 1$	0.71	0.2
$1 \Rightarrow 2$	0.71	0.2
$2 \Rightarrow 7$	0.29	0.5
$7 \Rightarrow 2$	0.8	0.5
$1 \Rightarrow 13$	0.21	0.2
$13 \Rightarrow 1$	0.75	0.2
$2 \Rightarrow 13$	0.29	1.2
$13 \Rightarrow 2$	1	4
$2 \Rightarrow 16$	0.14	0.6
$16 \Rightarrow 2$	1	0.6
$4 \Rightarrow 1$	0.63	-0.6
$1 \Rightarrow 4$	0.36	-0.6
$4 \Rightarrow 2$	0.5	-1.6
$2 \Rightarrow 4$	0.29	-1.6
$6 \Rightarrow 1$	0.67	-0.1
$1 \Rightarrow 6$	0.14	-0.1
$6 \Rightarrow 2$	1	0.9
$2 \Rightarrow 6$	0.21	0.9
$6 \Rightarrow 7$	0.67	1.25
$7 \Rightarrow 6$	0.4	1.25
$1, 7 \Rightarrow 2$	0.8	0.5
$1, 2 \Rightarrow 7$	0.4	1.5
$2, 7 \Rightarrow 1$	1	1.2
$1, 13 \Rightarrow 2$	1	0.9
$1, 2 \Rightarrow 13$	0.3	1
$2, 13 \Rightarrow 1$	0.75	0.2

Setelah dihilangkan *rule* yang mempunyai nilai Interestingness kurang dari Interestingness minimum yang ditetapkan kemudian di dapatkan *rule* yang akan digunakan untuk rekomendasi. *Rule* untuk rekomendasi ini dapat bdilihat pada tabel 3.22.

Tabel 3.22 Rule Rekomendasi

Rule	Conf	IR
1=>7	0.36	1.5
7=>1	1	1.5
2=>7	0.29	0.5
7=>2	0.8	0.5
2=>13	0.29	1.2
13=>2	1	4
2=>16	0.14	0.6
16=>2	1	0.6
6=>2	1	0.9
2=>6	0.21	0.9
6=>7	0.67	1.25
7=>6	0.4	1.25
1,7=>2	0.8	0.5
1,2=>7	0.4	1.5
2,7=>1	1	1.2
1,13=>2	1	0.9
1,2=>13	0.3	1

3.6 Perancangan Evaluasi

Skenario evaluasi algoritma MibARM adalah dengan menggunakan dataset transaksi sejumlah 500 transaksi dan 1000 transaksi. Minimum Support (0.01, 0.02, 0.03, 0.04, 0.05), minimum Confidence (0.2, 0.4, 0.6) dan minimum Interestingness (0.5, 1, 5, 10). Skenario untuk Evaluasi terdapat dalam tabel 3.23.

Tabel 3.23 Perancangan Evaluasi

Jumlah Transaksi	Sup _{min}	Conf _{min}	Imin	Jumlah kandiati (Ck)	Rule	Lift Rasio

UNIVERSITAS BRAWIJAYA

