BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem adalah suatu proses untuk mengumpulkan dan melakukan sebuah interpretasi pada kenyataan – kenyataan yang ada, mendiagnosa persoalan dan menggunakan keduanya untuk untuk memperbaiki sistem. Analisis Sistem ini bertujuan untuk menjelaskan permasalahan yang muncul pada saat pembangunan sistem, hal ini bertujuan untuk membantu ketika proses perancangan sistem berlangsung. Tugas utama analisis sistem dalam tahap ini adalah menemukan kelemahan-kelemahan dari sistem yang berjalan sehingga dapat diusulkan perbaikannya. Dalam analisis sistem ini meliputi beberapa bagian, yaitu :

1. Analisis Masalah
2. Analisis Penerepan Metode CRISP-DM
3. Analisis Penerapan Kebutuhan Perangkat Lunak
4. Analisis Kebutuhan Fungsional
5. Analisis Kebutuhan Non-Fungsional

3.1.1 Analisis Masalah

Analisis masalah yang dilakukan adalah melihat dan mengidentifikasi permasalahan atau kendala di dalam penelitian yang dilakukan. Masalah dapat diartikan sebagai sebagai suatu pernyataan yang menyatakan tentang situasi yang memerlukan pemecahan. Perlu ada analisis masalah melalui rumusan masalah yang sudah ditentukan sebelumnya. Berdasarkan hasil pengamatan, dapat disimpulkan bahwa permasalahan yang ada di Minimarket Warga Tunggal Bandung yaitu belum adanya informasi penentuan paket produk yang sesuai dengan minat konsumen yang akan ditawarkan menjelang hari-hari besar kegamaan.

3.1.2 Analisis Penerapan Metode CRISP-DM

Metode pembangunan perangkat *data mining* yang digunakan dalam penelitian ini adalah *Cross-Industry Standard Process for Data Mining* (CRISP-DM).

3.1.2.1 Pemahaman Bisnis

Tahapan pemahaman bisnis merupakan tahapan pertama dilakukan dalam kerangka kerja CRISP-DM. Dalam tahapan bisnis ini terdapat beberapa tahapan lainnya, yaitu :

1. Tujuan Bisnis

   Dalam proses bisnisnya Minimarket Warga Tunggal Bandung mempunyai tujuan bisnis yaitu untuk memenuhi permintaan konsumen sehingga dapat meningkatkan kepuasan pelanggan dan pendapatan Minimarket.

2. Penentuan Sasaran *Data Mining*

   Tujuan dari penerapan *data mining* pada pemaketan produk-produk ini adalah untuk mengetahui bagaimana pola pembelian kosumen dalam pemaketan produk dan menigkatkan jumlah produk yang dibeli dalam bentuk paket.

3.1.2.2 Pemahaman Data

Berdasarkan tujuan bisnis Minimarket Warga Tunggal Bandung, data dalam penelitian ini adalah data transaksi yang terjadi di Minimarket Warga Tungal Bandung selama 6 bulan yang berasal dari *file microsoft excel* (.xlsx), Data tersebut akan dilakukan proses *mining* pada program. Data yang akan dipakai dalam proses analisis ini merupakan data sample transaksi untuk percobaan dalam proses pencarian rule nya. Berikut merupakan penjelasan dari masing-masing *atribut* yang terdapat pada tabel data transaksi yang dapat dilihat pada tabel 3.1 dibawah ini :

Tabel 3. 1Struktur Data Transaksi

Keterangan		
Fungsi	Untuk mengetahui jenis produk yang paling diminati oleh konsumen	
Format	Microsoft Excel (.xlsx)	
Atribut	Id	Nomor urut
	Faktur	Kode transaksi dalam satu transaksi
	Kode barang	Kode setiap barang
	Nama barang	Nama barang yang dibeli
	Harga satuan	Harga setiap barang
	Qty	Jumlah barang yang dibeli
	Harga	Harga barang yang dibeli
	Total	Total harga barang yang dibeli
	Kategori	Kategori Barang yang dibeli
	Tanggal	Tanggal pencetakan struk
	Buyer	Pembeli
	Kasir	Nama kasir
	Kbuyer	Komputer tempat pelanggan melakukan pembayaran

3.1.2.3 Persiapan Data

Sebelum dilakukan proses *mining* data harus dipersiapkan sedemikian rupa agar data digunakan sesuai kebutuhan. Dalam kasus ini tidak semua *atribut* dalam data dipakai untuk membuat proses *data mining* lebih sederhana, sesuai kebutuhan dan mencapai tujuan yang diharapkan. Adapun tahapan-tahapan persiapan data dalam penelitian ini adalah sebagai berikut :

1. Pemilihan Atribut

   Berdasarkan informasi yang ingin didapat oleh pengguna mengenai barang - barang yang dibeli oleh konsumen, maka dalam tahap ini atribut yang digunakan dari tabel C-1 dalam Lampiran C hanya Faktur dan Kode Barang yang memiliki tujuan masing – masing, seperti Faktur berguna

untuk membedakan satu transaksi dengan transaksi lain, sedangkan Kode Barang berguna untuk mengetahui barang apa saja yang dibeli dalam satu transaksi. Hasil pemilihan atribut dapat dilihat pada tabel D-2 dalam Lampiran D.

2. Pembersihan Data

   Pada tahap Pembersihan Data akan dilakukan pengeleminasian terhadap data transaksi yang memiliki *item* tunggal karena data transaksi yang memiliki item tunggal tidak memiliki hubungan asosiasi dengan *item* lainnya . Hasil pembersihan data dapat dilihat pada tabel D-3 pada Lampiran D.

3.1.2.4 Pemodelan

Pemodelan merupakan tahap untuk membuat model atau desain sistem yang akan dibangun. *Data* yang digunakan untuk pengolahan *data* ini adalah *data* yang terahir yang sudah dilakukan proses pembersihan data dalam Lampiran D. Didalam tahap ini akan dilakukan penyelesaian masalah pada pemaketan produk di Minimarket Warga Tunggal Bandung dengan Metode *Association Rule* dan *Algoritma FP-Growth*. Langkah awal yang dilakukan dalam pemodelan ini adalah mencari kombinasi item yang memenuhi syarat dari nilai *minimum support* dalam database. *Minimum Support* merupakan suatu ukuran yang menunjukan seberapa besar tingkat dominasi suatu item pada sebuah data transaksi. Untuk kasus ini diambil nilai *minimum support* yang diambil adalah 12 yang berarti item yang memiliki jumlah *frequent* yang kurang dari 12 akan dieliminasi dalam data transaksi. Langkah pertama yang dilakukan untuk menghitung nilai *Support* dari sebuah item adalah dengan menghitung kemunculan tiap item yang sudah disusun secara *Ascending*. Berikut ini merupakan jumlah kemunculan tiap item :

Tabel 3. 2 Frequent Item

Kode Barang	Support Count
RK4	20
MIN78	19
RK6	18
SNC27	17
RK5	16
KP2	15
MIN4	15
RK2	15
MINUMAN112	5
OBT4	4
MAN1	4
MIN30	4
MIN25	4
MAN81	4
SNC14	3
MIN7	3
MAN3	2
PBT4	2
KP20	2
MAN79	1
SBN3	1
SBN8	1
SNC56	1
SNC77	1

Setelah mengetahui nilai *frequent* dari semua item, langkah selanjutnya adalah mengeliminasi item yang tidak memenuhi *minimum support* yang telah ditentukan. Hasil pengeliminasian item yang tidak memenuhi *minimum support* dapat dilihat pada Tabel 3.4

Tabel 3. 3 Item yang memenuhi Minimum Support

Kode Barang	Support Count
RK4	20
MIN78	19
RK6	18
SNC27	17
RK5	16
KP2	15
MIN4	15
RK2	15

Setelah didapatkan data yang memenuhi nilai *minimum support* seperti pada tabel 3.4 diatas, selanjutnya mencari *Frequency Of Occurrence*. Urutkan tabel 3.4 berdasarkan *Frequent* terbesar sampai dengan terkecil, kemudian berikan nilai *priority* dimulai dari data frekuensi terbesar sampai dengan terkecil. Jika ada Kode Barang dengan jumlah kemunculan yang sama maka yang didahulukan adalah Kode Barang yang pertama muncul dalam data transaksi. Hasil pemberian nilai priority dapat dilihat di Tabel 3.5

Tabel 3. 4 Frequency Of Occurence

Kode Barang	Support Count	Priority
RK4	20	1
MIN78	19	2
RK6	18	3
SNC27	17	4
RK5	16	5
KP2	15	6
MIN4	15	7
RK2	15	8

Setelah tiap item diketahui nilai *priority* nya maka data transaksi yang mengandung item yang memenuhi *minimum support* akan diurutkan berdasarkan nilai *priority*. Pada tahap ini item yang memiliki *frequent* tertinggi atau nilai *priority* tertinggi akan didahulukan penulisannya. Hasil pengurutan berdasarkan *priority* dapat dilihat pada Tabel 3.6 berikut ini :

Tabel 3. 5 Hasil Pengurutan Berdasarkan Priority

Faktur	Kode Barang
1000	RK4
1000	RK6
1000	SNC27
1000	RK5
1000	KP2
1000	MIN4
1000	RK2
1001	MIN78
1001	SNC27
1002	RK4
1002	MIN78
1002	RK6
1002	MIN4
1002	RK2
1003	MIN78
1003	RK6
1003	SNC27
1003	RK5
1003	KP2
1003	MIN4
1003	RK2
1004	RK4
1004	RK6
1004	RK5
1004	KP2
1005	MIN78

Faktur	Kode Barang
1005	RK6
1005	SNC27
1005	MIN4
1005	RK2
1006	RK4
1006	MIN78
1006	RK6
1006	SNC27
1006	RK5
1006	KP2
1007	RK4
1007	MIN78
1007	RK5
1007	RK2
1008	RK4
1008	RK6
1008	SNC27
1008	KP2
1008	MIN4
1009	RK4
1009	MIN78
1009	RK5
1010	RK4
1010	RK6

Faktur	Kode Barang
1010	SNC27
1010	MIN4
1011	MIN78
1011	RK6
1011	RK5
1011	KP2
1011	RK2
1012	RK4
1012	SNC27
1012	RK2
1013	MIN78
1013	RK6
1013	RK5
1013	KP2
1013	MIN4
1013	RK2
1014	RK4
1014	MIN78
1014	SNC27
1014	RK5
1014	KP2
1014	RK2
1015	RK4
1015	MIN78

Faktur	Kode Barang
1015	RK6
1015	RK5
1015	MIN4
1016	RK6
1016	SNC27
1016	KP2
1017	RK4
1017	MIN78
1017	SNC27
1017	KP2
1017	MIN4
1018	RK4
1018	RK5
1018	MIN4
1018	RK2
1019	RK4
1019	MIN78
1019	RK6
1019	SNC27
1019	RK5
1019	KP2
1020	MIN78
1020	RK6
1020	SNC27
1021	RK4

Faktur	Kode Barang
1021	MIN78
1021	RK6
1021	SNC27
1021	RK5
1021	KP2
1021	RK2
1022	MIN4
1022	RK2
1023	RK4
1023	MIN78
1023	SNC27
1023	KP2
1023	RK2
1024	RK4
1024	MIN78
1024	RK6
1024	RK5
1024	KP2
1025	RK4
1025	MIN4
1026	RK6
1026	RK5
1026	MIN4
1026	RK2
1027	RK4

Faktur	Kode Barang
1027	MIN78
1027	SNC27
1027	KP2
1027	RK2
1028	RK6
1028	MIN4
1029	RK4
1029	MIN78
1029	SNC27
1029	RK5
1029	MIN4

Setelah data sudah terurut berdasarkan *priority*, langkah selanjutnya adalah membangun *FP-Tree.* Berikut ini merupakan tahap – tahap dalam membangun *FP-Tree* :

a. Buat *Tree* dengan *Root* yang tidak memiliki anak.
b. Jika *Root* tidak memiliki anak, maka data yang muncul pertama kali dijadikan sebagai anak pertama dari *Root* disertai dengan *support count* = 1.
c. Jika *Root* sudah memiliki anak, tetapi data yang muncul berbeda dengan anak dari *Root*, maka data tersebut merupakan anak selanjutnya dari *Root* tersebut dan *support count* nya bertambah 1.
d. Jika *Root* sudah memiliki anak, dan data yang muncul sama dengan anak tersebut maka *support count*-nya bertambah 1.
e. Jika data yang muncul sudah ada dalam notasi anak, maka *support count*-nya bertambah 1.

Hasil pembentukan *FP-Tree* tiap transaksi pada data transaksi ini dapat dilihat di Lampiran C-4 pada Lampiran C.

1. Setelah tahap pembangunan *fp-tree* dari sekumpulan data transaksi dilakukan, maka akan dilakukan proses untuk mencari *frequent itemset* yang signifikan. Berikut adalah tahapanya :
   a) Tahap Pembangkitan Conditional Pattern Base

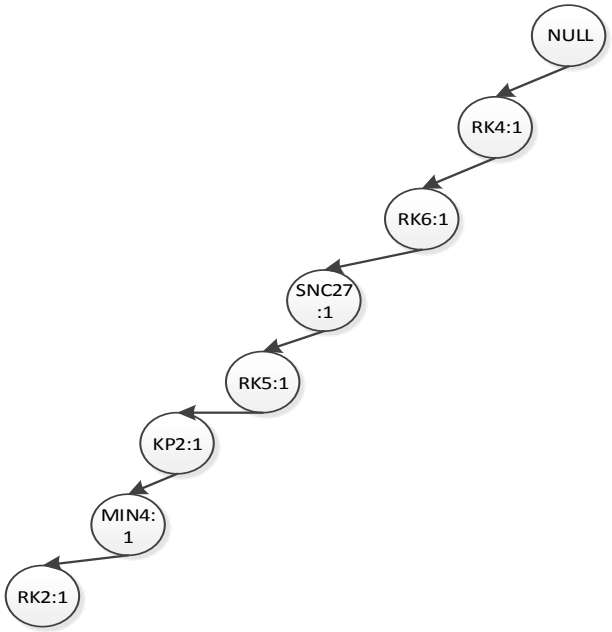
      Pembangkitan *Conditional Pattern Base* didapatkan melalui hasil *FP-Tree* seluruhnya dengan mencari *support count* terkecil sesuai dengan hasil pengurutan *priority* yang telah dibuat sebelumnya.
   b) Tahap Pembangkitan Conditional FP-Tree

      Pada tahap ini, *support count* dari setiap item pada Tabel *Conditional Pattern Base* diatas dijumlahkan, lalu setiap item yang memiliki jumlah *Support Count* lebih besar atau sama dengan *Minimum Support* akan dibangkitkan dengan *Conditional FP-Tree.*
   c) Tahap Pencarian *Frequent Itemset*

      Apabila *Conditional FP-Tree* merupakan lintasan tunggal (*single path),* maka didapatkan *frequent itemset* dengan melakukan kombinasi

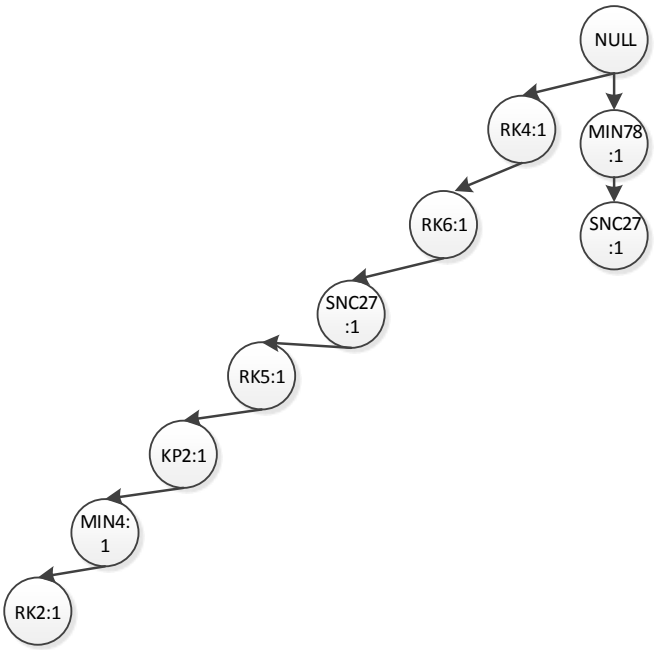
*item*untuk setiap *conditional* FP-*Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan secara rekursif.

Berikut ini merupakan pembentukan FP-Tree untuk Faktur 1000 dengan produk yang dipesan adalah RK4,RK6,SNC27,RK5,KP2,MIN4,RK2 :



Gambar 3. 1 Hasil pembentukan FP-Tree Faktur 1000

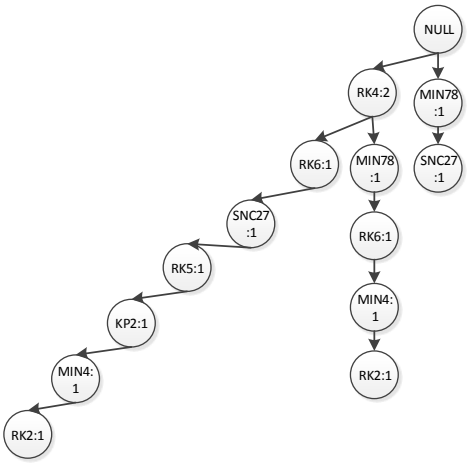
Pembacaan Faktur 1001 yaitu MIN78,SNC27. Karena MIN78 belum terbentuk maka dibuat jalur baru seperti pada Gambar 3.2 :



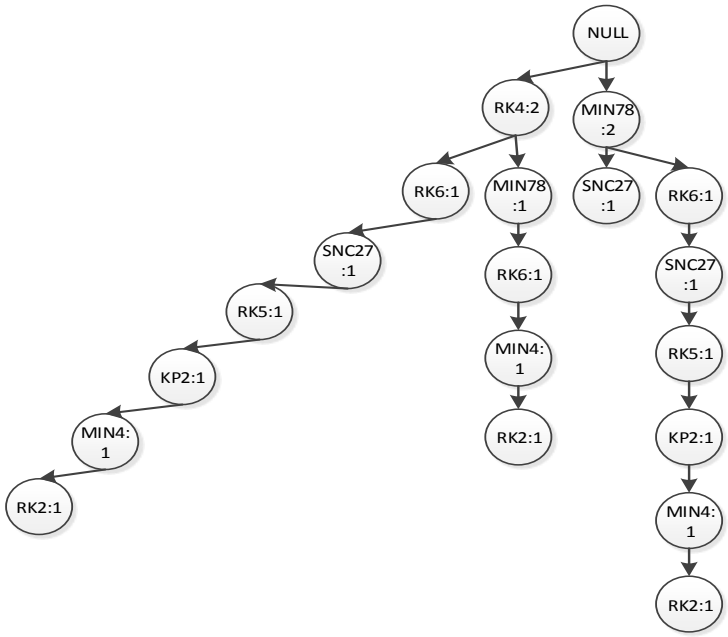
Gambar 3. 2 Hasil pembentukan FP-Tree Faktur 1001

Berikut ini merupakan hasil pembentukan FP-Tree Faktur 1002 dengan barang yang dipesan adalah RK4,MIN78,RK6,MIN4,RK2 :

Berikut ini merupakan hasil pembentukan FP-Tree Faktur 1003 dengan barang yang dipesan adalah MIN78,RK6,SNC27,RK5,KP2,MIN4,RK2:

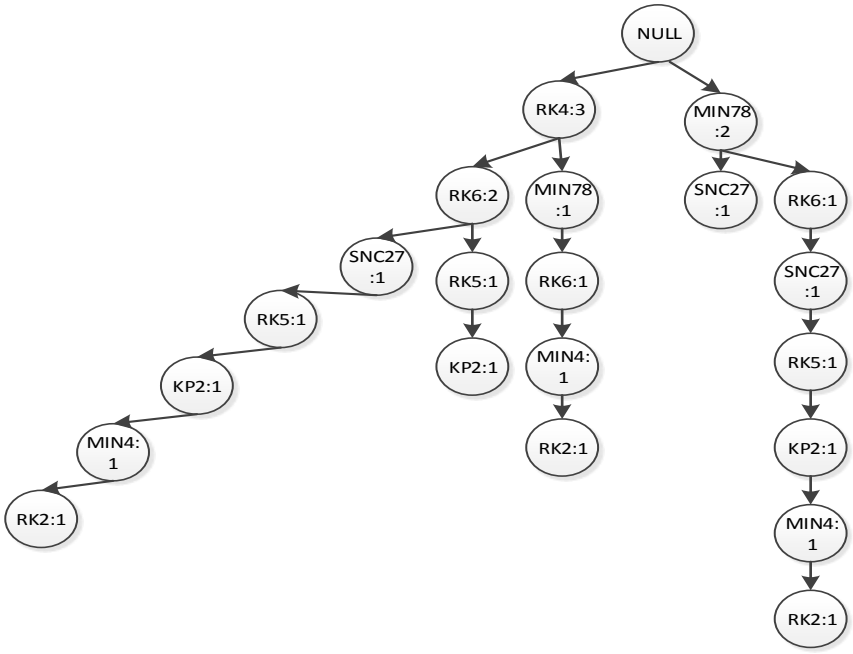


Gambar 3. 3Hasil pembentukan FP-Tree Faktur 1002



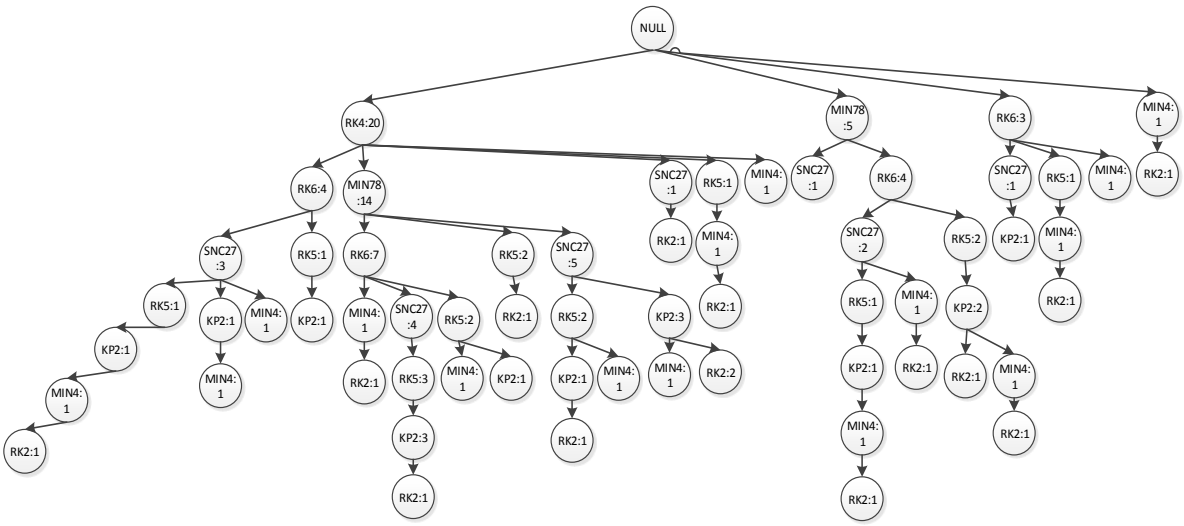
Gambar 3. 4 Hasil pembentukan FP-Tree Faktur 1003

Berikut ini merupakan hasil pembentukan FP-Tree Faktur 1004 dengan barang yang dipesan adalah RK4,RK6,RK5,KP2 :



Gambar 3. 5 Hasil pembentukan FP-Tree Faktur 1004

Setelah proses pembacaan Faktur 1029 yaitu RK4,MIN78,SNC27,RK5,MIN4, maka terbentuklah *Tree* dari semua transaksi yang akan digunakan seperti pada gambar 3.6 dibawah ini :



Gambar 3. 6 Hasil pembentukan FP-Tree Faktur 1029

Setelah tahap pembangunan *FP-Tree* dari sekumpulan data transaksi dilakukan, maka akan diterapkan *Algoritma FP-Growth* untuk proses pencarian *frequent itemset* yang signifikan. Berikut langkah-langkah utama *Algoritma FP-Growth*, yaitu :

a. Tahap pembangkitan *conditional pattern base*

   Pembangkitan *conditional pattern base* didapatkan melalui hasil FP-Tree seluruhnya, dengan mencari *support count* terkecil sesuai dengan hasil pengurutan *priority.* Dan telah didapatkan yaitu : RK2, MIN4, KP2, RK5, SNC27, RK6, MIN78, RK4. Berikut adalah hasil tabel dari *conditional pattern base :*

Tabel 3. 6 Conditional Pattern Base

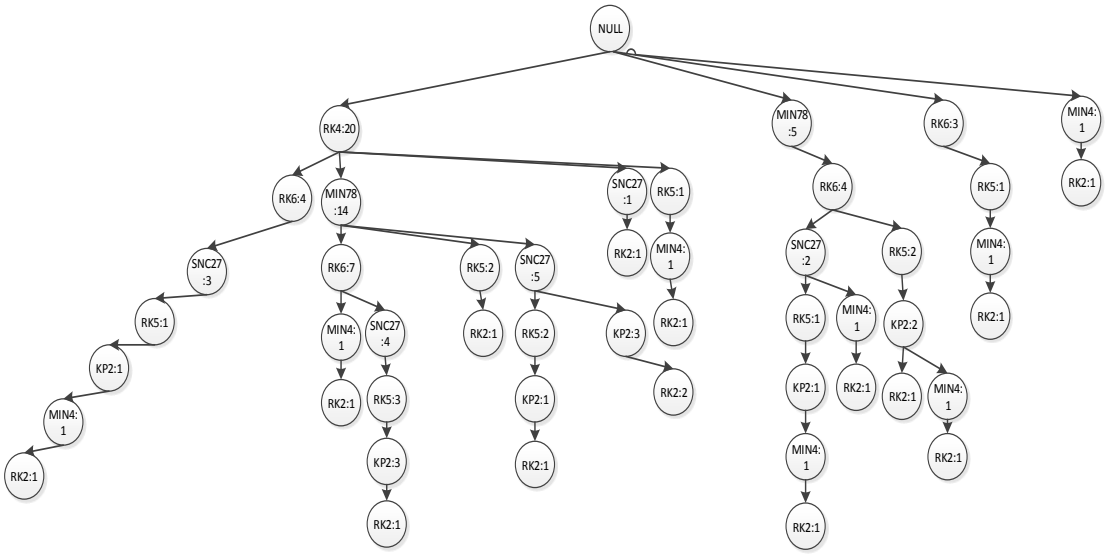
Item	*Conditonal Pattern Base*
RK2	{{RK4,RK6,SNC27,RK5,KP2,MIN4:1},{RK4,MIN78,RK6,MIN4:1}, {RK4,MIN78,RK6,SNC27,RK5,KP2:1},{RK4,MIN78,RK5:1},{RK4,MIN78, SNC27,RK5,KP2:1},{RK4,MIN78,SNC27,KP2:1}, {RK4,SNC27:1},{RK4,RK5,MIN4:1},{MIN78,RK6,SNC27,RK5,KP2,MIN4: 1},{MIN78,RK6,SNC27,MIN4:1},{MIN78,RK6,RK5,KP2:1}, {MIN78,RK6,RK5,KP2,MIN4:1},{RK6,RK5,MIN4:1},{MIN4:1}}
MIN4	{{RK4,RK6,SNC27,RK5,KP2:1},{RK4,RK6,SNC27,KP2:1}, {RK4,RK6,SNC27:1},{RK4,MIN78,RK6:1},{RK4,MIN78,RK6,RK5:1},{RK 4,MIN78,SNC27,RK5:1},{RK4,MIN78,SNC27,KP2:1}, {RK4,RK5:1},{RK4:1},{MIN78,RK6,SNC27,RK5,KP2:1}, {MIN78,RK6,SNC27:1},{MIN78,RK6,RK5,KP2:1},{RK6,RK5:1}, {RK6:1}}
KP2	{{RK4,RK6,SNC27,RK5:1},{RK4,RK6,SNC27:1},{RK4,RK6,RK5:1},{RK4 ,MIN78,RK6,SNC27,RK5:3},{RK4,MIN78,RK6,RK5:1}, {RK4,MIN78,SNC27,RK5:1},{RK4,MIN78,SNC27:3},{MIN78,RK6,SNC27, RK5:1},{MIN78,RK6,RK5:2},{RK6,SNC27:1}}
RK5	{{RK4,RK6,SNC27:1},{RK4,RK6:1},{RK4,MIN78,RK6,SNC27:3}, {RK4,MIN78,RK6:2},{RK4,MIN78:2},{RK4,MIN78,SNC27:2}, {RK4:1},{MIN78,RK6,SNC27:1},{MIN78,RK6:2},{RK6:1}}
SNC27	{{RK4,RK6:3},{RK4,MIN78,RK6:4},{RK4,MIN78:5}, {RK4,MIN78:5},{MIN78:1},{MIN78,RK6:2},{RK6:1}}
RK6	{{RK4:4},{RK4,MIN78:7},{MIN78:4},{NULL, RK6:3}}
MIN78	{{RK4,MIN78:14},{NULL, MIN78:5}}
RK4	{NULL, RK4:20}

b. Tahap pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan *minimumsupport* akan dibangkitkan dengan *ConditionalFP-Tree.*

- *ConditionalFP-Tree* untuk kode produk RK2 :

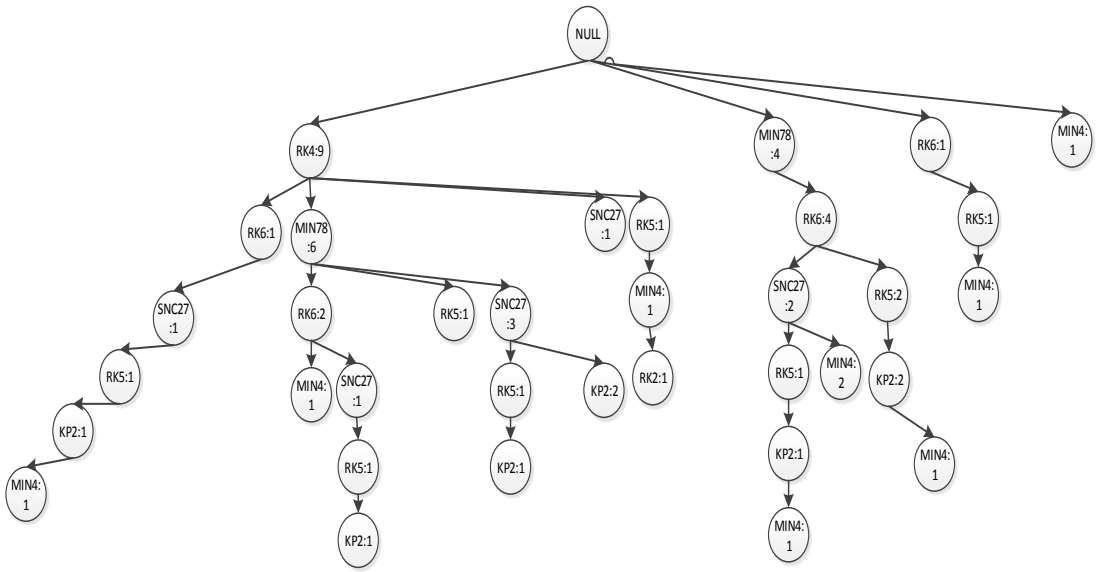
{RK4,RK6,SNC27,RK5,KP2,MIN4:1},{RK4,MIN78,RK6,MIN4:1},{RK4,MIN78,RK6,SNC27,RK5,KP2:1},{RK4,MIN78,RK5:1},{RK4,MIN78,SNC27,RK5,KP2:1},{RK4,MIN78,SNC27,KP2:2},{RK4,SNC27:1},{RK4,RK5,MIN4:1},{MIN78,RK6,SNC27,RK5,KP2,MIN4:1},{MIN78,RK6,SNC27,MIN4:1},{MIN78,RK6,RK5,KP2:1},{MIN78,RK6,RK5,KP2,MIN4:1},{RK6,RK5,MIN4:1},{MIN4:1}



Gambar 3. 7 Conditional FP-Tree Kode Produk RK2 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk RK2 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk RK2 akan dibangkitkan. Berikut adalah tahapannya :

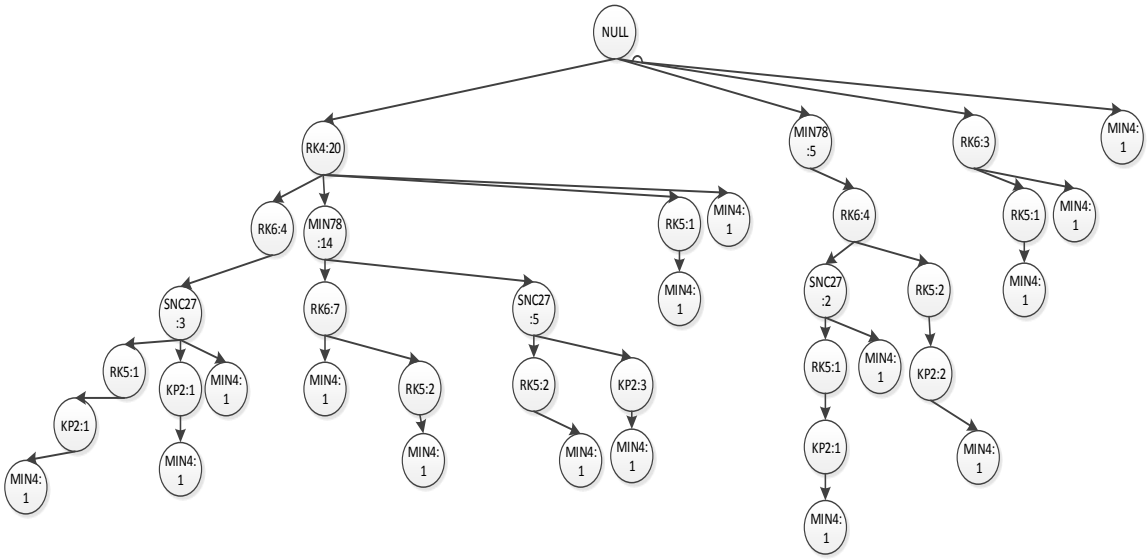
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Pada lintasan yang berakhiran {RK2:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran RK2, akan disamakan dengan *support count* RK2 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 8 Conditional FP-Tree Kode Produk RK2 Setelah dibangkitkan

Tidak ada *frequent itemset* yang keluar karna tidak ada yang memenuhi *minimum support.*

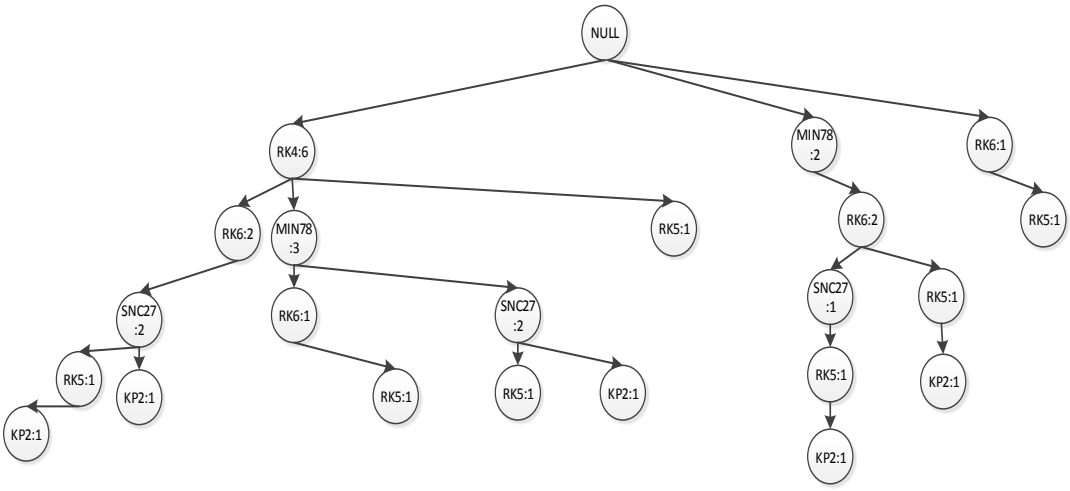
- *ConditionalFP-Tree* untuk kode produk MIN4 :
  {RK4,RK6,SNC27,RK5,KP2:1},{RK4,RK6,SNC27,KP2:2},{RK4,RK6,SNC27:1},{RK4,MIN78,RK6:1},
  {RK4,MIN78,RK6,RK5:1},{RK4,MIN78,SNC27,RK5:1},
  {RK4,MIN78,SNC27,KP2:1},{RK4,RK5:1},{RK4:1},
  {MIN78,RK6,SNC27,RK5,KP2:1},{MIN78,RK6,SNC27:1},
  {MIN78,RK6,RK5,KP2:1},{RK6,RK5:1},{RK6:1},{ NULL, MIN4:1}



Gambar 3. 9 Conditional FP-Tree Kode Produk MIN4 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk MIN4 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk MIN4 akan dibangkitkan. Berikut adalah tahapannya :

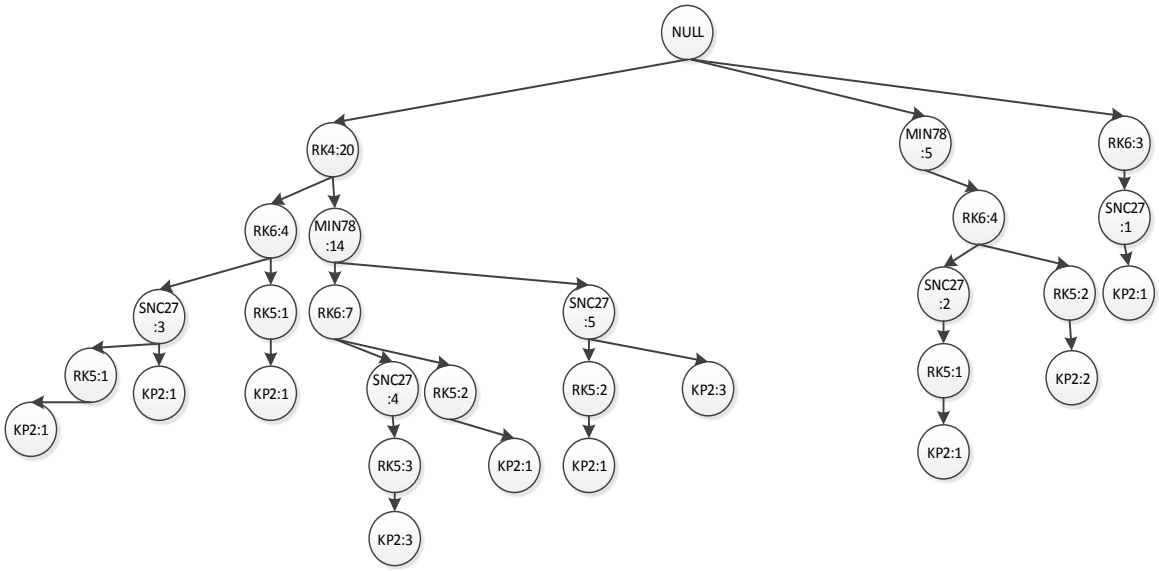
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran {MIN4:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran MIN4, akan disamakan dengan *support count* MIN4 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 10 Conditional FP-Tree Kode Produk MIN4 Setelah dibangkitkan

Tidak ada *frequent itemset* yang keluar karena tidak ada yang memenuhi *minimum support.*

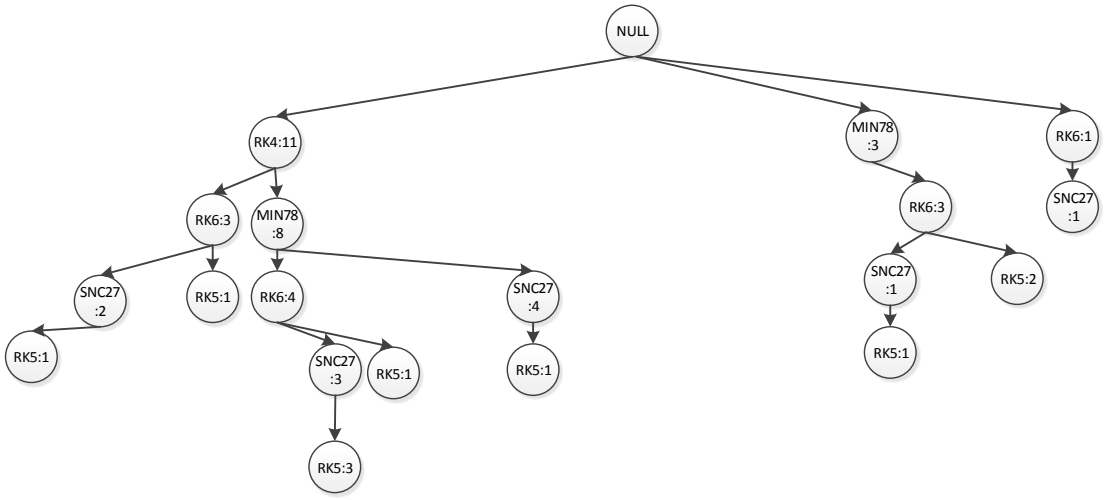
- *ConditionalFP-Tree* untuk kode produk KP2 : {RK4,RK6,SNC27,RK5:1},{RK4,RK6,SNC27:2},{RK4,RK6,RK5:1}, {RK4,MIN78,RK6,SNC27,RK5:3},{RK4,MIN78,RK6,RK5:1}, {RK4,MIN78,SNC27,RK5:1},{RK4,MIN78,SNC27:1},{MIN78,RK6,SNC27,RK5:1}, {MIN78,RK6,RK5:2}, {RK6,SNC27:1}



Gambar 3. 11 Conditional FP-Tree Kode Produk KP2 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk KP2 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk KP2 akan dibangkitkan. Berikut adalah tahapannya :

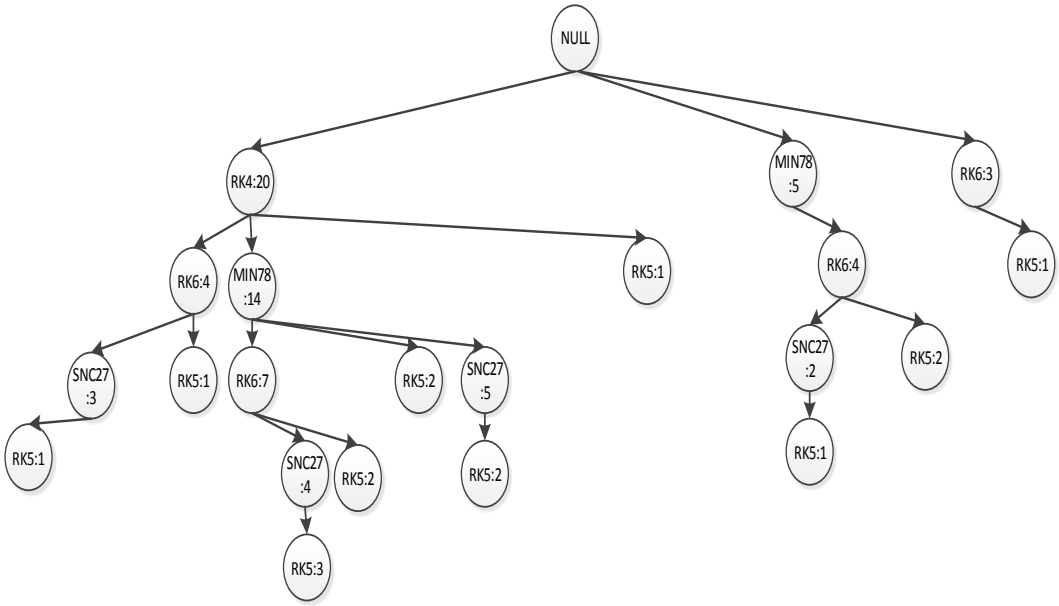
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran {KP2:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran KP2, akan disamakan dengan *support count* KP2 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 12 Conditional FP-Tree Kode Produk KP2 Setelah dibangkitkan

Tidak ada *frequent itemset* yang keluar karena tidak ada yang memenuhi *minimum support.*

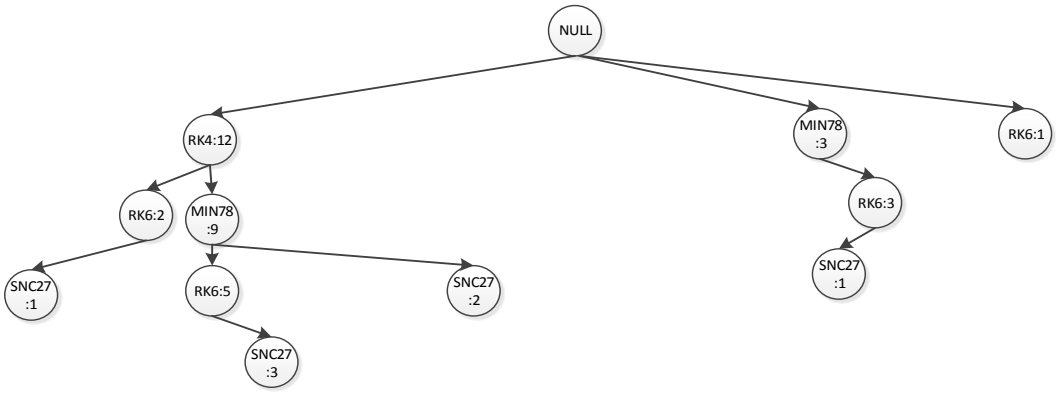
- *ConditionalFP-Tree* untuk kode produk RK5 : {RK4,RK6,SNC27:1},{RK4,RK6:1},{RK4,MIN78,RK6,SNC27:3},{RK4, MIN78,RK6:2},{RK4,MIN78:2},{RK4,MIN78,SNC27:2},{RK4:1},{MIN 78,RK6,SNC27:1},{MIN78,RK6:2},{RK6:1}



Gambar 3. 13 Conditional FP-Tree Kode Produk RK5 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk RK5 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk RK5 akan dibangkitkan. Berikut adalah tahapannya :

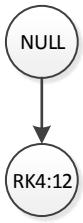
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran { RK5:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran RK5, akan disamakan dengan *support count*RK5 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



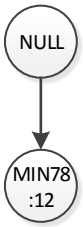
Gambar 3. 14 Conditional FP-Tree Kode Produk RK5 Setelah dibangkitkan

Frequent Itemset yang keluar adalah {RK4:12}, {MIN78:12}

Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. {RK4:12} {MIN78:12}adalah yang keluar yang dapat memenuhi *minimum support*. Dan *frequentitemset* yang didapatkan adalah {RK5, RK4:12} dan {RK5, MIN78:12}

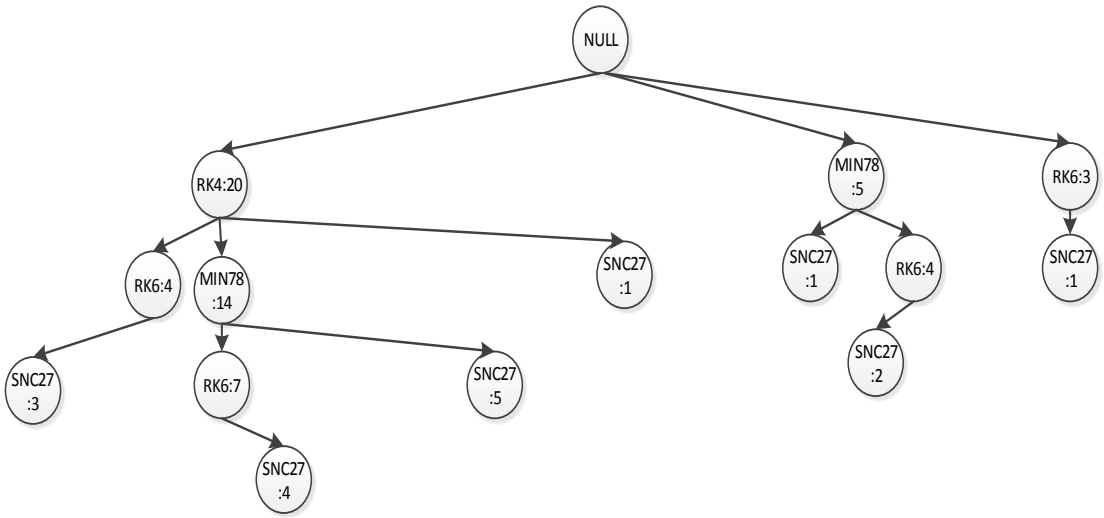


Gambar 3. 15 Conditional FP-Tree RK5, RK4:12 Hasil Frequent Pattern



Gambar 3. 16 Conditional FP-Tree RK5, MIN78:12 Hasil Frequent Pattern

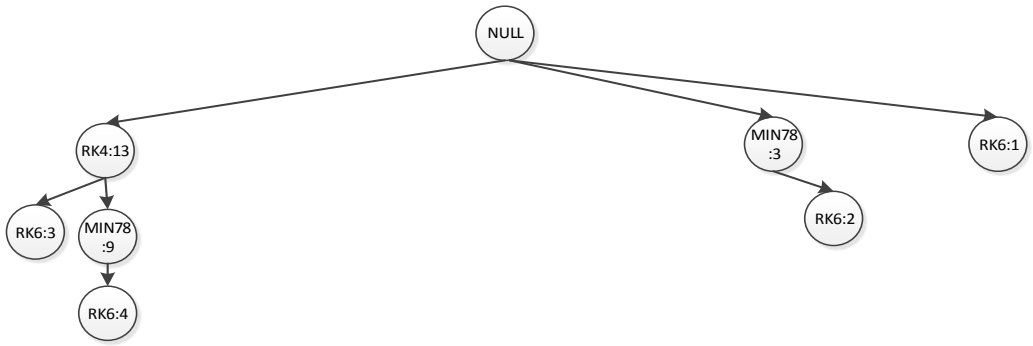
ConditionalFP-Tree untuk kode produk SNC27 :
{RK4,RK6:3},{RK4,MIN78,RK6:4},{RK4,MIN78:5},{RK4:1},{MIN78:1},{MIN78,RK6:2},{RK6:1}



Gambar 3. 17 Conditional FP-Tree Kode Produk SNC27 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk SNC27 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk SNC27 akan dibangkitkan. Berikut adalah tahapannya :

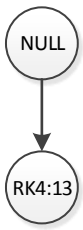
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran { SNC27:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran SNC27, akan disamakan dengan *support count*SNC27 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



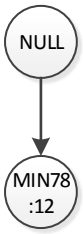
Gambar 3. 18 Conditional FP-Tree Kode Produk SNC27 Setelah dibangkitkan

frequent itemset yang keluar adalah {RK4:13}, {MIN78:12}

Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. {RK4:13} {MIN78:12}adalah yang keluar yang dapat memenuhi *minimum support.* Dan *frequentitemset* yang didapatkan adalah {SNC27, RK4:12} dan {SNC27, MIN78:12}

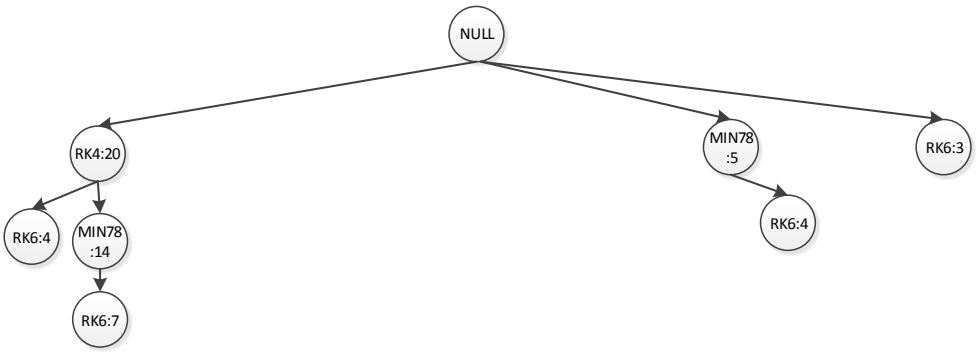


Gambar 3. 19 Conditional FP-Tree SNC27, RK4:13 Hasil Frequent Pattern



Gambar 3. 20 Conditional FP-Tree SNC27, MIN78:12 Hasil Frequent Pattern

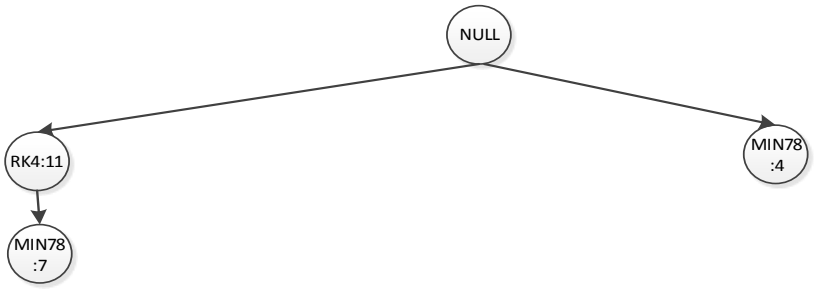
- *ConditionalFP-Tree* untuk kode produk RK6 :
  {RK4:4},{RK4,MIN78:7},{MIN78:4},{NULL,RK6:3}



Gambar 3. 21 Conditional FP-Tree Kode Produk RK6 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk RK6 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk RK6 akan dibangkitkan. Berikut adalah tahapannya :

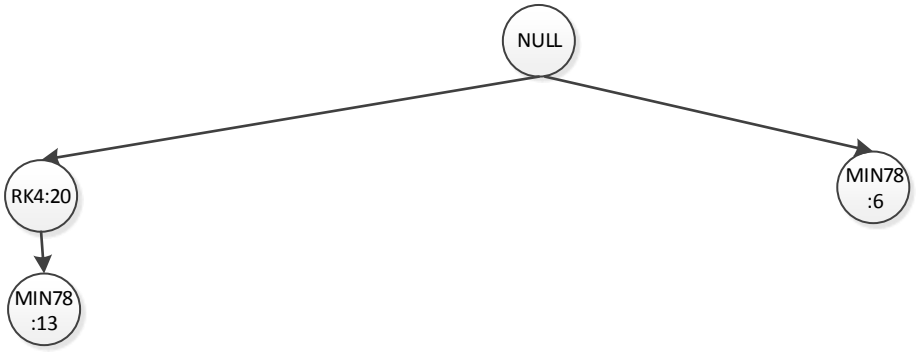
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran { RK6:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran RK6, akan disamakan dengan *support count*RK6 yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 22 Conditional FP-Tree Kode Produk RK6 Setelah dibangkitkan

Tidak ada *frequent itemset* yang keluar karna tidak ada yang memenuhi *minimum support.*

- *ConditionalFP-Tree* untuk kode produk MIN78 :
  {RK4,MIN78:14} {NULL, MIN78:5}



Gambar 3. 23 Conditional FP-Tree Kode Produk MIN78 Sebelum dibangkitkan

Karena nilai *Support Count*kode produk MIN78 setelah diakumulasikan memenuhi *Minimum Support*, maka kode produk MIN78 akan dibangkitkan. Berikut adalah tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional FP-Tree*.
   - Pada lintasan yang berakhiran { MIN78:1} dengan *support count* adalah 1 atau lebih akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran MIN78, akan disamakan dengan *support count*MIN78 yaitu sesuai dengan *support count* yang dimilikinya.

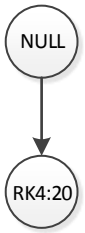
- Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 24 Conditional FP-Tree Kode Produk MIN78 Setelah dibangkitkan

frequent itemset yang keluar adalah {RK4:13}

- *ConditionalFP-Tree* untuk kode produk RK4 :
  {NULL, RK4:20}



Gambar 3. 25 Conditional FP-Tree Kode Produk RK4 Sebelum dibangkitkan

Setelah mendapatkan hasil dari *conditional fp-tree,* tidak ditemukan beberapa *frequent itemset* yang keluar seperti pada table dibawah ini :

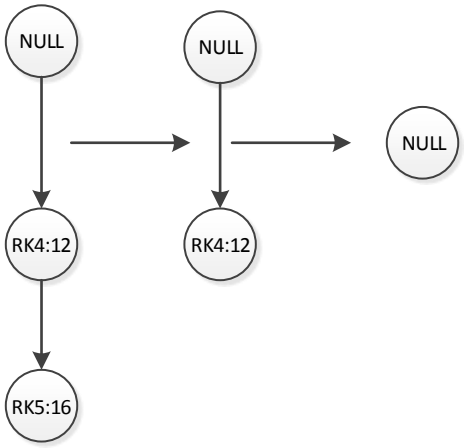
Tabel 3. 7 Conditional Fp-Tree

Item	***Conditional FP-tree***
RK5	<RK5, RK4:12>,<RK5, MIN78:12>
SNC27	<SNC27, RK4:13>,<SNC27, MIN78:12>
MIN78	<MIN78, RK4:13>

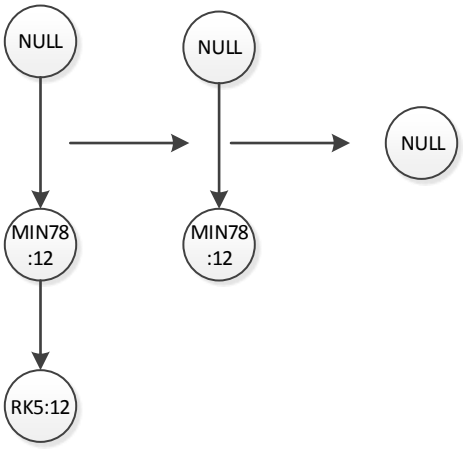
Setelah mendapatkan tabel dari *conditional fp-tree*, tahap selanjutnya adalah :

a. Akumulasikan setiap *item* untuk setiap *frequent itemset* yang ada pada table 3.6
b. Buat *conditional fp-tree* dari setiap *frequent itemset* yang keluar.

- Untuk item RK5, item frequent itemset yang keluar adalah {RK4:12} {MIN78:12}

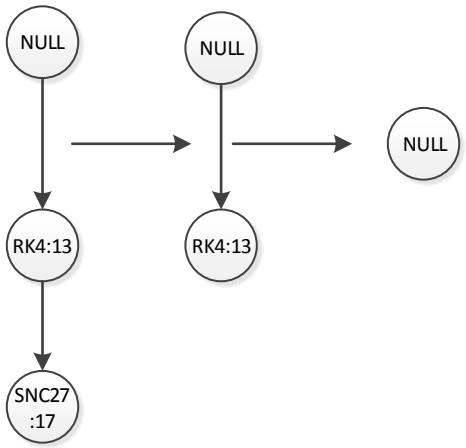


Gambar 3. 26 Conditional FP-Tree Untuk Item Frequent itemset RK5, RK4:12

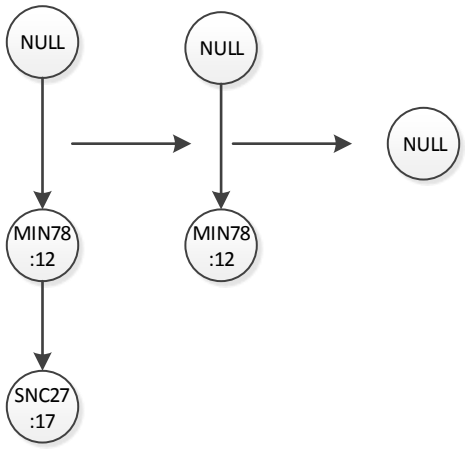


Gambar 3. 27 Conditional FP-Tree Untuk Item Frequent itemset RK5, MIN78:12

- Untuk item SNC27, item frequent itemset yang keluar adalah {RK4:12} {MIN78:12}

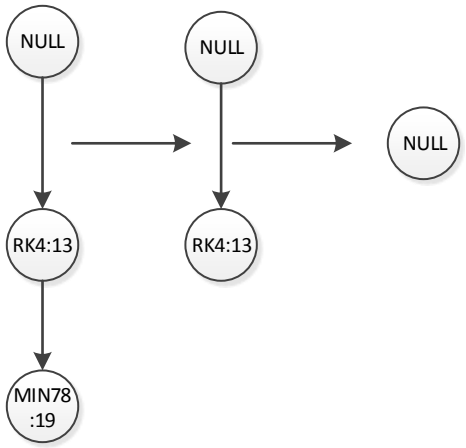


Gambar 3. 28 Conditional FP-Tree Untuk Item Frequent itemset SNC27, RK4:12



Gambar 3. 29 Conditional FP-Tree Untuk Item Frequent itemset SNC27, MIN78:1

- Untuk item MIN78, item frequent itemset yang keluar adalah {RK4:13}



Gambar 3. 30 Conditional FP-Tree Untuk Item Frequent itemset MIN78, RK4:13

Setelah mendapatkan hasil *conditional fp tree* dari tiap *frequent itemset*, selanjutnya adalah menentukan *suffix* dari tiap *frequent itemset*, seperti pada table dibawah ini :

Tabel 3. 8 Hasil Frequent Itemset

Suffix	***Frequent Itemset***
RK5	RK5 - RK4 RK5 - MIN78
SNC27	SNC27 - RK4 SNC27 - MIN78
MIN78	MIN78 - RK4

Dari hasil *frequent itemset* yang didapatkan pada table 3.8 di atas akan di *generate* untuk mendapatkan *rule* dengan *minimum support* 12 dan *minimum confidence* 60 % dengan masing-masing *itemset* dikombinasikan dengan *itemset* lain. Dengan menggunakan rumus yang ada pada bagian bab 2 untuk memperoleh nilai *support* dan nilai *confidence*.Berikut hasil perhitungan *minimum support* dan *minimum confidence*.

Tabel 3. 9 Hasil Perhitungan Confidence

Rule	***Minimum Confidence***
RK5 → RK4	(12/16)*100%=75%
RK4 → RK5	(12/20)*100%=60%
RK5 → MIN78	(12/16)*100%= 75%
MIN78 → RK5	(12/19)* 100 %=63,15%
SNC27 → RK4	(12/17) *100% =70,59%
RK4 → SNC27	(12/20) *100% =60%
SNC27→ MIN78	(12/17) *100% =70,59%
MIN78→ SNC27	(12/19) *100 %=63,15%
MIN78 → RK4	(13/19) *100% =68,42%
RK4 → MIN78	(13/20) *100% =65%

Berikut adalah hasil *rule* dalam penentuan barang-barang dalam bentuk awal berdasarkan *confidence*

Tabel 3. 10 Paket Produk dalam Bentuk Awal

Rule	***Confidence***
RK5 → RK4	75%
RK4 → RK5	60%
RK5 → MIN78	75%
MIN78 → RK5	63,15%
SNC27 → RK4	70,59%
RK4 → SNC27	60%
SNC27 → MIN78	70,59%
MIN78 → SNC27	63,15%
MIN78 → RK4	68,42%
RK4 → MIN78	65%

Tabel 3. 11 Hasil Pembentukan Rules

Rule	**Paket Produk**
RK5 → RK4	Jika 234 (16) dibeli, maka SAMPURNA KRETEK dibeli dengan confidence 75%
RK4 → RK5	Jika SAMPURNA KRETEK dibeli, maka 234 (16) dibeli dengan confidence 60%
RK5 → MIN78	Jika 234 (16) dibeli, maka JAM GELAS dibeli dengan confidence 75%
MIN78 → RK5	Jika JAM GELAS dibeli, maka 234 (16) dibeli dengan confidence 63,15%
SNC27 → RK4	Jika ROMA KELAPA dibeli, maka SAMPURNA KRETEK dibeli dengan confidence 70,59%
RK4 → SNC27	Jika SAMPURNA KRETEK dibeli, maka ROMA KELAPA dibeli dengan confidence60%
SNC27 → MIN78	Jika ROMA KELAPA dibeli, maka JAM GELAS dibeli dengan confidence 70,59%
MIN78 → SNC27	Jika JAM GELAS dibeli, maka ROMA KELAPA dibeli dengan confidence 63,15%

MIN78 → RK4	Jika JAM GELAS dibeli, maka SAMPURNA KRETEK dibeli dengan confidence 68,42%
RK4 → MIN78	Jika SAMPURNA KRETEK dibeli, maka JAM GELAS dibeli dengan confidence 65%

Dalam penentuan kombinasi produk yang akan dijual dalam bentuk paket pihak Minimarket Warga Tunggal tidak memiliki batasan khusus, sehingga dari Rule yang terbentuk bisa digunakan sebagai pertimbangan dalam menentukan paket produk yang akan dijual. Berikut ini merupakan paket produk yang terbentuk.

Tabel 3. 12 Informasi Paket

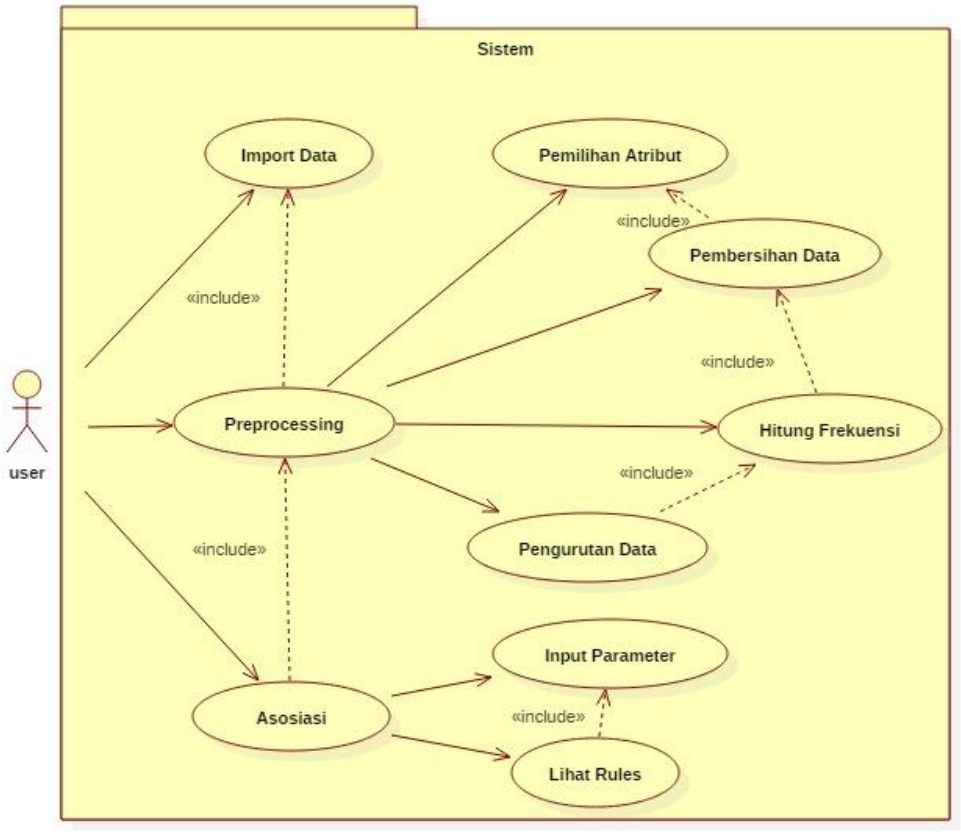
Paket	**Paket Produk**
Paket 1	234 (16) bisa dipaketkan dengan SAMPURNA KRETEK
Paket 2	Jika 234 (16) bisa dipaketkan dengan JAM GELAS
Paket 3	Jika ROMA KELAPA bisa dipaketkan dengan SAMPURNA KRETEK
Paket 4	Jika ROMA KELAPA bisa dipaketkan dengan JAM GELAS
Paket 5	JAM GELAS bisa dipaketkan dengan SAMPURNA KRETEK

3.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk perancangan terhadap aplikasi yang akan dibangun. Aplikasi yang akan dibangun menggunakan pendekar berorientasi objek dengan menggunakan pemodelan UML. Pemodelan yang akan digunakan untuk memodelkan terdiri dari diagram *Use Case*, A*ctivity Diagram, Sequence Diagram* dan *Class Diagram.*

3.2.1 Diagram Use Case

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 9*use case*. Adapun diagram *use case* dari sistem yang akan dibuat dapat dilihat di gambar 3.1.



Gambar 3. 31Diagram Use Case Pada Sistem Data Mining di Minimarket Warga Tunggal

Tabel 3. 13 Tabel Definisi User

User	**Deskripsi**
User	*User* bertugas untuk melakukan *import* data, memasukkan *Minimum Support* dan *Minimum Confidence*

Tabel 3. 14 Deskripsi Use Case

No	Use Case	Deskripsi
1	*Import* Data	*Import* data harus dilakukan user terlebih dahulu, data yang di*import*merupakan data transaksi yang selanjutnya data tersebut akan disimpankedalam *database*
2	*Preprocessing*	Data transaksi yang sudah tersimpan dalam *database* kemudian akan dilakukan proses pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan berdasarkan *priority*
3	Pemilihan Atribut	Sistem melakukan pemilihan atribut Faktur dan Kode Barang, proses ini hanya bisa dilakukan jika *user* sudah melakukan *import* data
4	Pembersihan Data	Sistem melakukan pembersihan data pada Faktur yang mengandung Kode Barang tunggal, proses ini hanya bisa dilakukan jika user sudah melakukan *import* data dan pemilihan atribut
5	Hitung *frekuensi*	Sistem melakukan proses perhitungan kemunculan tiap item pada data transaksi
6	Pengurutan Data	Sistem melakukan proses pengurutan data berdasarkan *priority.*
7	Asosiasi	Sistem melakukan proses eliminasi *dan* lihat *rule*. Sedangkan *user* menginputkan nilai parameter yaitu nilai *MinimumSupport* dan *MinimumConfidence*.
8	Input Parameter	User memasukkan nilai *Minimum Support* dan *Minimum Confidence* sebagai parameter dalam menentukan *rule.*
9	Rules	*Sistem* menampilkan informasi dari hasil seleksi rule. Informasi yang ditampilkan berupa produk apa saja yang bisa dijual dalam bentuk paket.

3.2.1.1 *Skenario Use Case*

Skenario use case menjelaskan skenario dari setiap proses yang digambarkan pada diagram use case. Berikut ini skenariouse case dari gambar 3.1.

1. Skenario *Use Case Import Data*

Skenario *use case import* data menggambarkan langkah – langkah aksi *user* terhadap sistem untuk melakukan *import* data transaksi penjualan yang akan disimpan ke dalam database.

Tabel 3. 15 Requirement A.1

Requirement A.1
Sistem menyediakan menu*import data* untuk melakukan *import* data transaksi kedalam *database*sebelum melakukan *preprocessing* dan asosiasi

Tabel 3. 16 Skenario Use Case Import Data

Use case Name	***Import Data***	
Related Requirements	Requirement *A.1*	
Goal In Context	*Import* data transaksi penjualan dan menyimpannya ke dalam *database*	
Precondition	*User* menyiapkan data transaksi	
Successful End Condition	Data transaksi yang di *import* oleh *user* berhasil disimpan ke dalam *database*	
Failed End Condition	Data transaksi gagal disimpan ke dalam *database*	
Actors	*User*	
Triger	*User* memasukan data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*User* melakukan *request import* data
	2	*User* memasukkan data transaksi
	3	Sistem melakukan *generate file*
	4	Sistem menampilkan data di *gridview*
	5	Data disimpan ke *database*
	6	Sistem menampilkan pesan data tersimpan
Extension	Step	Branching Action
	6.1	Menampilkan pesan kesalahan dalam proses *import* data

2. Skenario *Use Case Preprocesing*

Skenario *use case preprocessing* menjelaskan langkah-langkah untuk melakukan proses Pemilihan Atribut, Pembersihan Data, Hitung Frekuensi dan Pengurutan Data.

Tabel 3. 17 Requirement A.2

Requirement A.2
Sistem menyediakan menu*preprocesing* untuk melakukan pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data.

Tabel 3. 18 Skenario Use Case Preprocesing

Use case Name	***Preprocesing***	
Related Requirements	Requirement *A.1*, Requirement *A.2*	
Goal In Context	Sistem melakukan proses pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data	
Precondition	Data transaksi sudah tersimpan didalam *database*	
Successful End Condition	Sistem berhasil melakukan proses pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data	
Failed End Condition	Sistem gagal untuk melakukan proses pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data	
Actors	*User*	
Triger	*User* melakukan *request* proses pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data kepada sistem	
Included Cases	Pemilihan Atribut, Pembersihan Data, Hitung Frekuensi dan Pengurutan Data	
Main Flow	Step	Action
	1	*User* melakukan *request preprocessing*
	2	*User* melakukan *request* Pemilihan Atribut
	3	Sistem melakukan proses pemilihan atribut
	4	User melakukan *request* pembersihan data
	5	Sistemmelakukan proses pembersihan data
	6	User melakukan *request* hitung frekuensi
	7	Sistem melakukan proses hitung frekuensi
	8	*User* melakukan *request* pengurutan data
	9	Sistem melakukan proses pengurutan data
Extension	Step	Branching Action
	-	-

3. Scenario *Use Case Pemilihan Atribut*

Skenario *Use Case* Pemilihan Atributmenggambarkan langkah-langkah melakukan pemilihan atribut terhadap data transaksi yang telah melewati proses *import data*.

Tabel 3. 19 Skenario Use Case Pemilihan Atribut

Use case Name	***Pemilihan Atribut***	
Related Requirements	Requirement *A.1*, Requirement *A.2*	
Goal In Context	Sistem melakukan pemilihan atribut Faktur dan Kode Barang	
Precondition	Data transaksi tersimpan didalam *database*	
Successful End Condition	Sistem berhasil memilih atribut Faktur dan Kode Barang	
Failed End Condition	Sistem gagal melakukan Pemilihan Atribut Faktur dan Kode Barang	
Actors	*User*	
Triger	*User* melakukan *request* Pemilihan Atribut kepada system	
Included Cases	-	
Main Flow	Step	Action
	1	*User request preprocessing*
	2	*User* melakukan *request* pemilihan atribut
	3	Sistem menampilkan hasil pemilihan atribut
	4	Sistem menampilkan pesan berhasil melakukan proses pemilihan atribut
Extension	Step	Branching Action
	4.1	Sistem manampilkan pesan gagal dalam melakukan proses pemilihan atribut

4. Skenario *Use Case Pembersihan Data*

Skenario *use casePembersihan Data*menggambarkan langkah-langkah melakukan pembersihan terhadap Fakturdan Kode Barang yang memiliki item tunggal dan telah melewati proses Pemilihan Atribut.

Tabel 3. 20 Skenario Use Case Pembersihan Data

Use case Name	***Pembersihan Data***

Related Requirements	Requirement *A.1,* Requirement *A.2*	
Goal In Context	Sistem melakukan pembersihan pada atribut Faktur dan Kode Barang yang memiliki item tunggal.	
Precondition	Data Transaksi sudah melalui proses Pemilihan Atribut	
Successful End Condition	Sistem berhasil melakukan pembersihan pada atribut Faktur dan Kode Barang yang memiliki item tunggal	
Failed End Condition	Sistem gagal dalam melakukan pembersihan pada atribut Faktur dan Kode Barang yang memiliki item tunggal	
Actors	*User*	
Triger	*User* melakukan *request* Pembersihan Data pada sistem	
Included Cases	-	
Main Flow	Step	Action
	1	*User* melakukan *request* pembersihan data
	2	Sistem menghapus data item tunggal
	3	Sistem menampilkan pesan berhasil melakukan proses pembersihan data
	4	Sistem menyimpan data hasil pembersihan data ke database
	5	Sistem menampilkan hasil proses pembersihan data
Extension	Step	Branching Action
	5.1	Sistem menampilkan pesan gagal menampilkan proses pembersihan data

5. Skenario *Use Case Hitung Frekuensi*

Skenario *Use Case*Hitung Frekuensimenggambarkan langkah-langkah melakukan perhitungan jumlah kemunculan tiap item dalam data transaksi dan telah melewati proses Pembersihan Data.

Tabel 3. 21 Skenario Use Case Hitung Frekuensi

Use case Name	**Hitung Frekuensi**
Related Requirements	Requirement *A.1,* Requirement *A.2*
Goal In Context	Sistem melakukan perhitungan kemunculan tiap item pada data transaksi .
Precondition	Data Transaksi sudah melalui proses Pembersihan Data
Successful End Condition	Sistem berhasil melakukan perhitungan jumlah kemunculan tiap item pada data transaksi
Failed End Condition	Sistem gagal dalam melakukan perhitungan jumlah kemunculan tiap item pada data transaksi

Actors	*User*	
Triger	*User* melakukan *request* Hitung Frekuensi pada sistem	
Included Cases	-	
Main Flow	Step	Action
	1	*User* melakukan *request* hitung frekuensi
	2	Sistem melakukan proses hitung frekuensi tiap item
	3	Sistem menampilkan pesan berhasil melakukan proses hitung frekuensi
	4	Sistem menampilkan hasil proses hitung frekuensi
Extension	Step	Branching Action
	4.1	Sistem menampilkan pesan gagal melakukan proses hitung frekuensi

6. Skenario *Use Case Pengurutan Data*

Skenario *Use Case*Pengurutan Datamenggambarkan langkah-langkah melakukan pengurutan data berdasarkan *priority*dan telah melewati proses Hitung Frekuensi .

Tabel 3. 22 Skenario Use Case Pengurutan Data

Use case Name	**Pengurutan Data**	
Related Requirements	Requirement *A.1,* Requirement *A.2*	
Goal In Context	Sistem melakukan pengurutan data berdasarkan *priority* .	
Precondition	Data Transaksi sudah melalui proses Hitung Frekuensi	
Successful End Condition	Sistem berhasil melakukan pengurutan data berdasarkan *priority*	
Failed End Condition	Sistem gagal dalam melakukan pengurutan data berdasarkan *priority*	
Actors	*User*	
Triger	*User* melakukan *request* Pengurutan Data pada sistem	
Included Cases	-	
Main Flow	Step	Action
	1	User melakukan request pengurutan data
	2	Sistem melakukan proses pengurutan data berdasarkan *priority*
	3	Sistem menampilkan pesan berhasil melakukan proses pengurutan data
	4	Sistem menampilkan hasil pengurutan data

Extension	Step	Branching Action
	4.1	Sistem menampilkan pesan gagal melakukan proses pengurutan data

7. Skenario Use Case Asosiasi

Skenario *Use Case Assosiasi* menggambarkan langkah-langkah aksi *user* untuk melakukan proses Input Parameterdan *Rules*

Tabel 3. 23 Requirement A.3

Requirement A.3
Sistem menyediakan menu asosiasi untuk melakukan input parameter berupa *Minimum Support* dan *Minimum Confidence*, Eliminasi untuk mengeliminasi item apasaja yang tidak memenuhi nilai parameter yang diinputkan dan Rules untuk melihat *rule* apa saja yang dihasilkan.

Tabel 3. 24 Skenario Use Case Asosiasi

Use case Name	**Asosiasi**	
Related Requirements	Requirement *A.1,* Requirement *A.2,* Requirement *A.3*	
Goal In Context	Sistem mendapatkan *rule* yang akan ditampilkan dalam bentuk informasi paket produk	
Precondition	Data sudah melalui proses *preprocessing*	
Successful End Condition	Sistem berhasil mendapatkan *rule* yang akan ditampilkan dalam bentuk informasi paket produk	
Failed End Condition	Sistem gagal mendapatkan *rule* yang akan ditampilkan dalam bentuk informasi paket produk	
Actors	*User*	
Triger	*User* menginputkan nilai parameter berupa nilai *Minimum Support* dan *Minimum Confidence*	
Included Cases	Input Parameter, Rules	
Main Flow	Step	Action
	1	*User* melakukan *request* asosiasi
	2	*User* memasukkan nilai parameter
	3	User melakukan *request* hasil analisa
	4	Sistem melakukan *validasi* inputan user
	5	Sistem melakukan pembentukan *rule*
	6	Sistem menampilkan hasil *rule*
Extension	Step	Branching Action

	4.1	Sistem menampilkan pesan nilai *MinimumSupport* yang dimasukkan tidak sesuai
	6.1	Sistem menampilkan pesan bahwa *rule* tidak ditemukan

8. Scenario *Use Case* Input Parameter

Skenario *use case* Input Parametermenggambarkan langkah-langkah aksi *user* untuk melakukan *input* nilai *Minimum Support* dan *Minimum Confidence* sebagai pendukung dalam pencarian *frequent itemset*.

Tabel 3. 25 Skenario Use Case Input Parameter

Use case Name	**Input Parameter**	
Related Requirements	Requirement *A.3*	
Goal In Context	Sistem mendapatkan nilai inputan *MinimumSupport* dan *Minimum Confidence* dari *user*	
Precondition	*User* memasukkan nilai *Minimum Support* dan *Minimum Confidence*	
Successful End Condition	Sistem berhasil mendapatkan nilai *MinimumSupport* dan *Minimum Confidence* sesuai dengan nilai yang dimasukkan *user*	
Failed End Condition	Sistem gagal mendapatkan nilai *MinimumSupport* dan *Minimum Confidence* dari *user*	
Actors	*User*	
Triger	*User* memasukkan nilai *MinimumSupport* dan *Minimum Confidence* yang sesuai	
Main Flow	Step	Action
	1	*User* melakukan *request* asosiasi
	2	User memasukkan nilai *minimum support* dan *minimum confidence*
	3	Sistem melakukan *validasi* inputan user
	4	Sistem menampilkan nilai *minimum support* dan *minimum confidence*
Extension	Step	Branching Action
	3.1	Sistem menampilkan pesan kesalahan nilai *Minimum Support* dan *Minimum Confidence* yang dimasukkan tidak sesuai

9. Skenario *Use Case* Rules

Skenario *Use Case Rules* menggambarkan langkah-langkah aksi *user* untuk melakukan proses *Rules*.

Tabel 3. 26 Skenario Use Case Rules

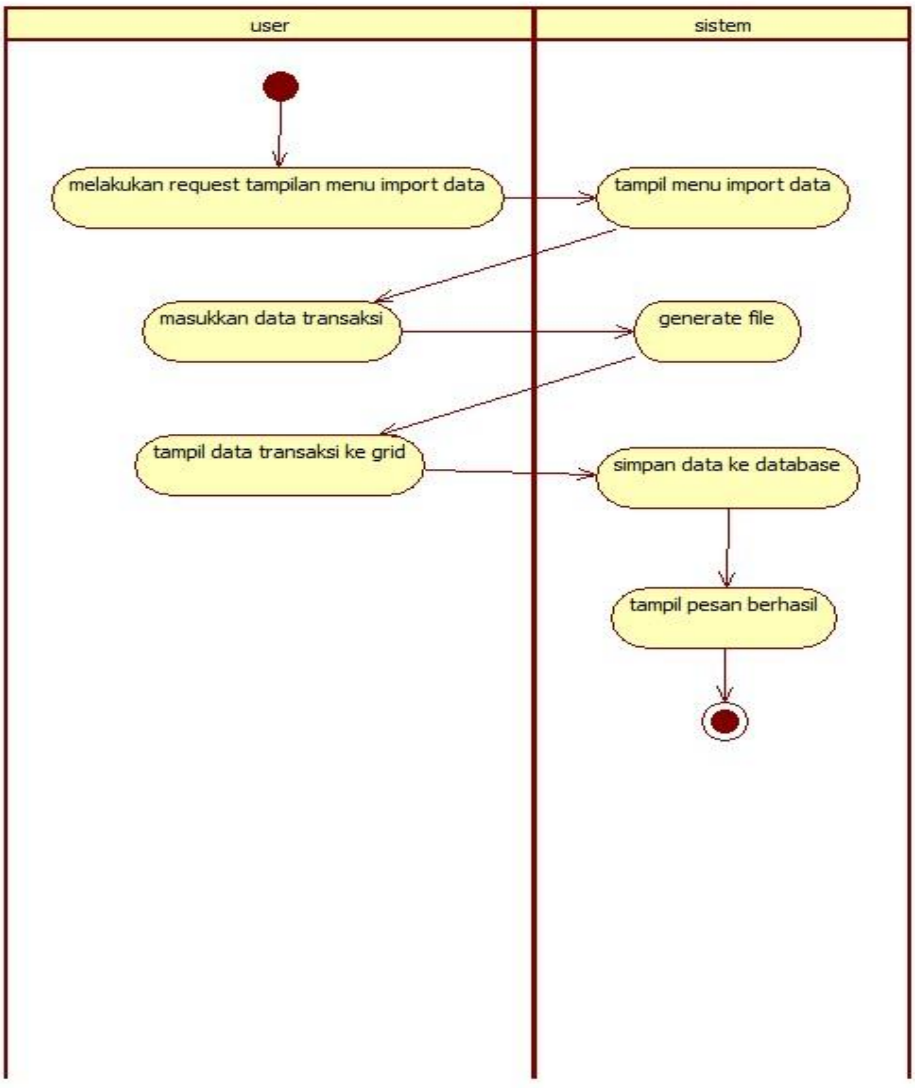
Use case Name	**Rules**	
Related Requirements	Requirement *A.1,* Requirement *A.2,* Requirement *A.3*	
Goal In Context	Sistem menampilkan informasi paket produk hasil dari pencarian *frequent itemset* dan pembentukan *rule*	
Precondition	User sudah melakukan memasukkan nilai *Minimum Support* dan *Minimum Confidence*	
Successful End Condition	Sistem berhasil menampilkan informasi paket produk hasil dari pencarian *frequent itemset* dan pembentukan *rule*	
Failed End Condition	Sistem gagal menampilkan informasi paket produk hasil dari pencarian *frequent itemset* dan pembentukan *rule*	
Actors	*User*	
Triger	*User* memasukkan nilai *MinimumSupport* dan *MinimumConfidence*	
Main Flow	Step	Action
	1	Sistem melakukan *request* data *confidence item*
	2	Sistem melakukan proses pembuatan Rules
	3	Sistem melakukan seleksi terhadap *item* yang memenuhi nilai parameter
	4	Sistem menyimpan hasil seleksi *item* ke *database*
	5	Sistem menampilkan hasil *rules*, nilai support dan confidence tiap item
	6	User melihat *rules* yang terbentuk
Extension	Step	Branching Action
	5.1	Sistem menampilkan pesan bahwa rule tidak ditemukan

3.2.2 Activity Diagram

Activity Diagram menggambarkan rangkaian aliran dari aktivitas dari setiap proses. Aktivitas diagram dari setiap proses dapat dilihat pada gambar-gambar berikut :

1. *Activity Diagram Import Data*

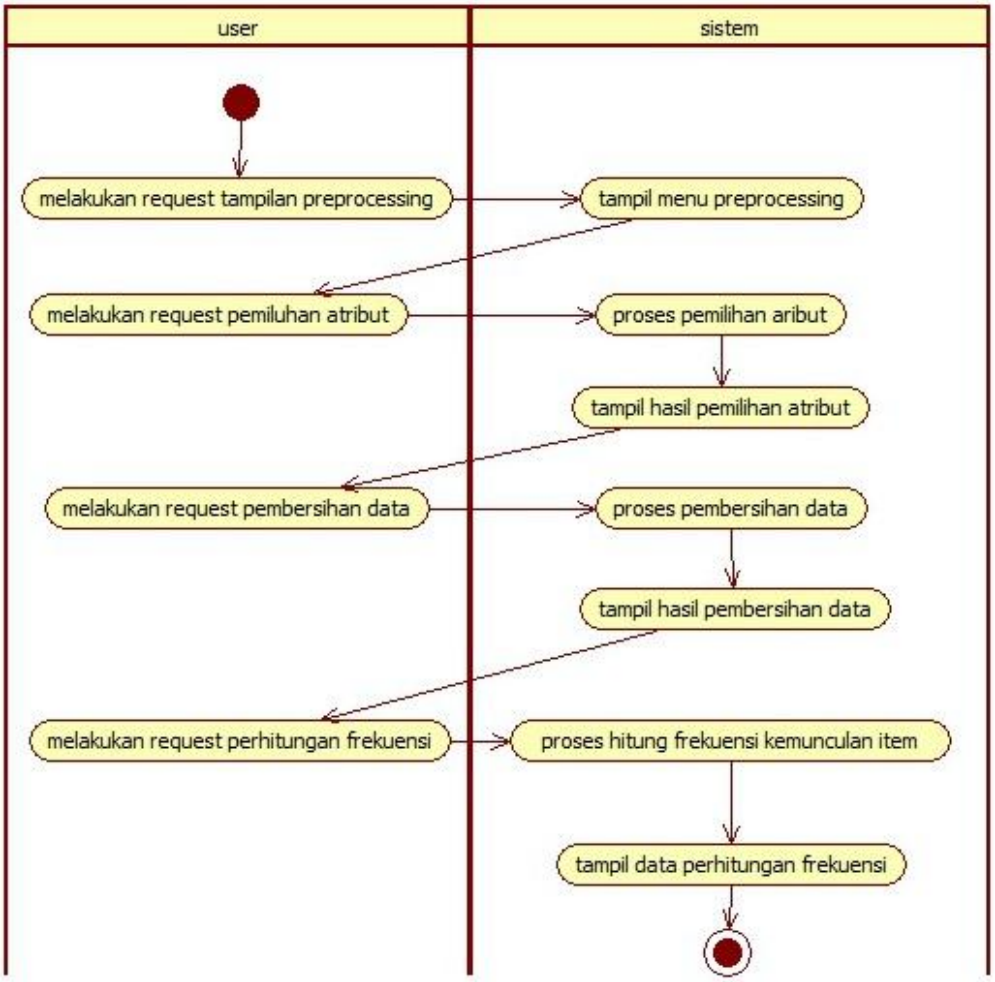
   Activity diagram import data menjelaskan aktivitas *user* dalam melakukan proses *import* data. *Activity diagramimport* data dapat dilihat pada gambar 3.2



Gambar 3. 32 Activity Diagram Import Data

2. *Activity Diagram Preprocesing*

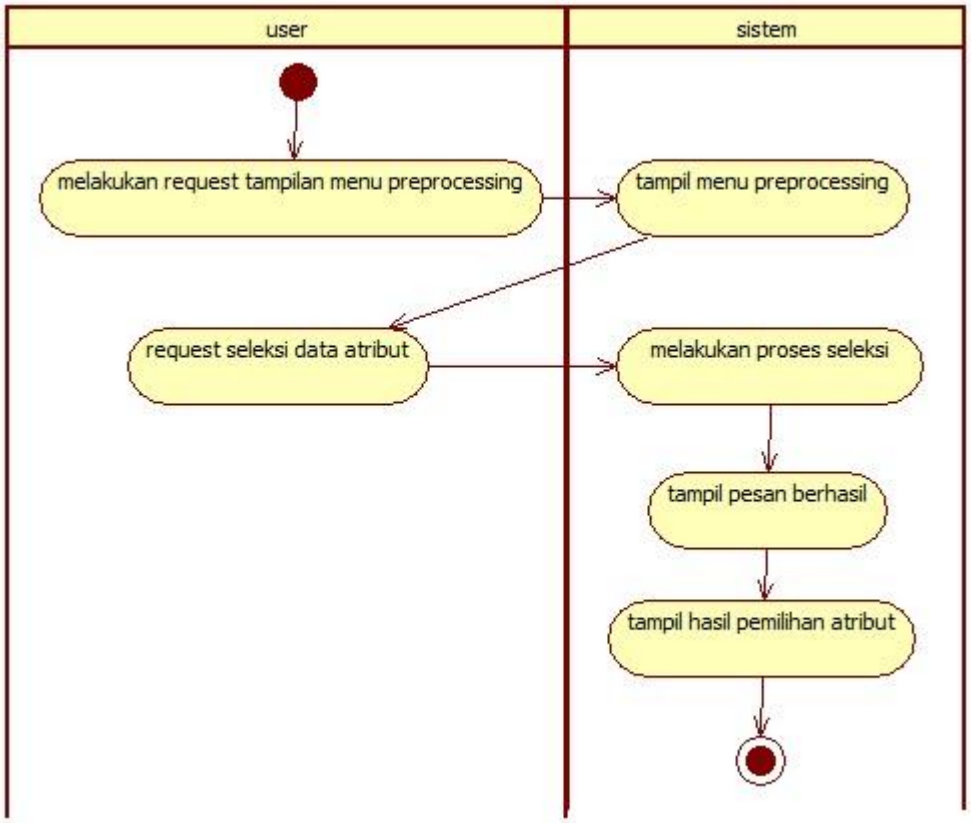
   Activity diagram preprocessing menjelaskan aktivitas sistem dalam melakukan proses Pemilihan Atributdan Pembersihan Data. *Activity diagrampreprocessing* dapat dilihat pada gambar 3.3



Gambar 3. 33 Activity Diagram Preprocessing

3. *Activity Diagram* Pemilihan Atribut

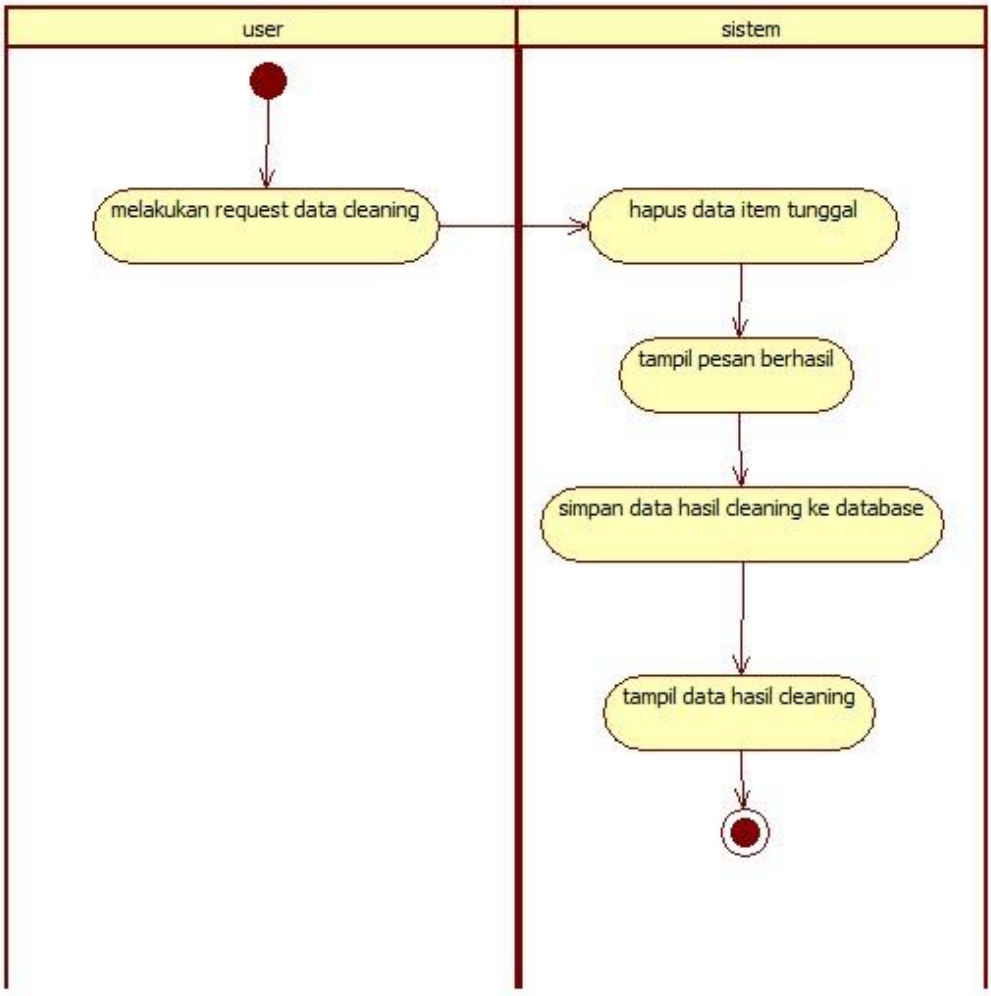
   Activity diagram Pemilihan Atribut menjelaskan aktivitas sistem dalam melakukan proses *data Pemilihan Atribut* . *Activity diagramdata Pemilihan Atribut* dapat dilihat pada gambar 3.4



Gambar 3. 34 Activity Diagram Data Pemilihan Atribut

4. *Activity Diagram Pembersihan Data*

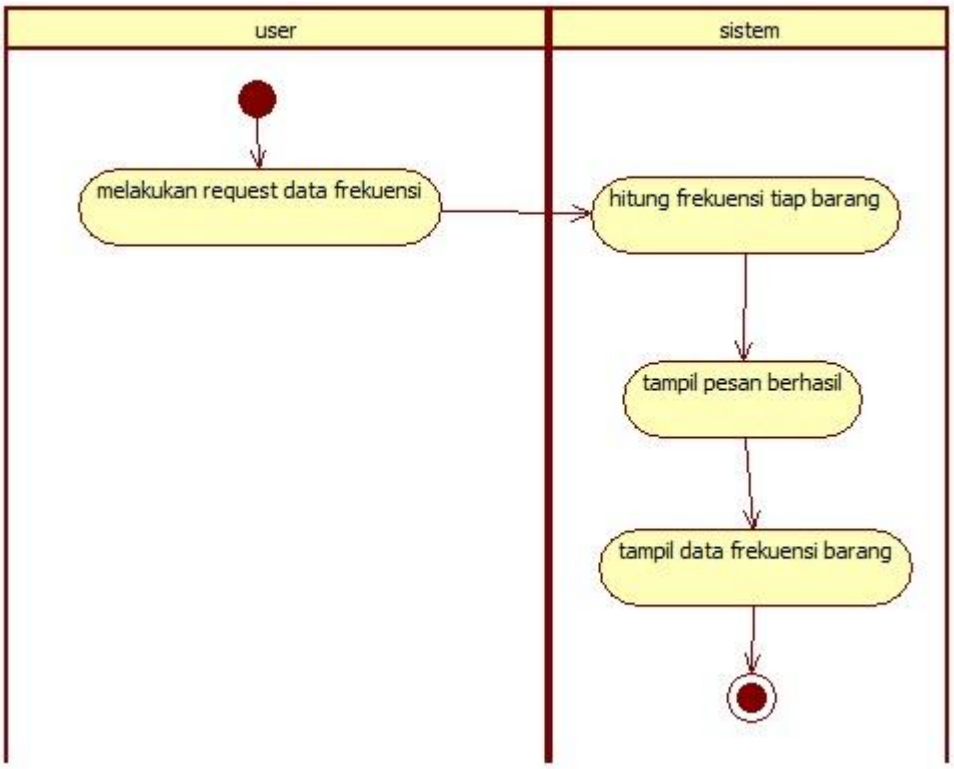
   Activity diagram data Pembersihan Data menjelaskan aktivitas sistem dalam melakukan proses *data Pembersihan Data*. *Activity diagramdata Pembersihan Data* dapat dilihat pada gambar 3.5



Gambar 3. 35 Activity Diagram Data Pembersihan Data

5. *Activity Diagram* Hitung Frekuensi

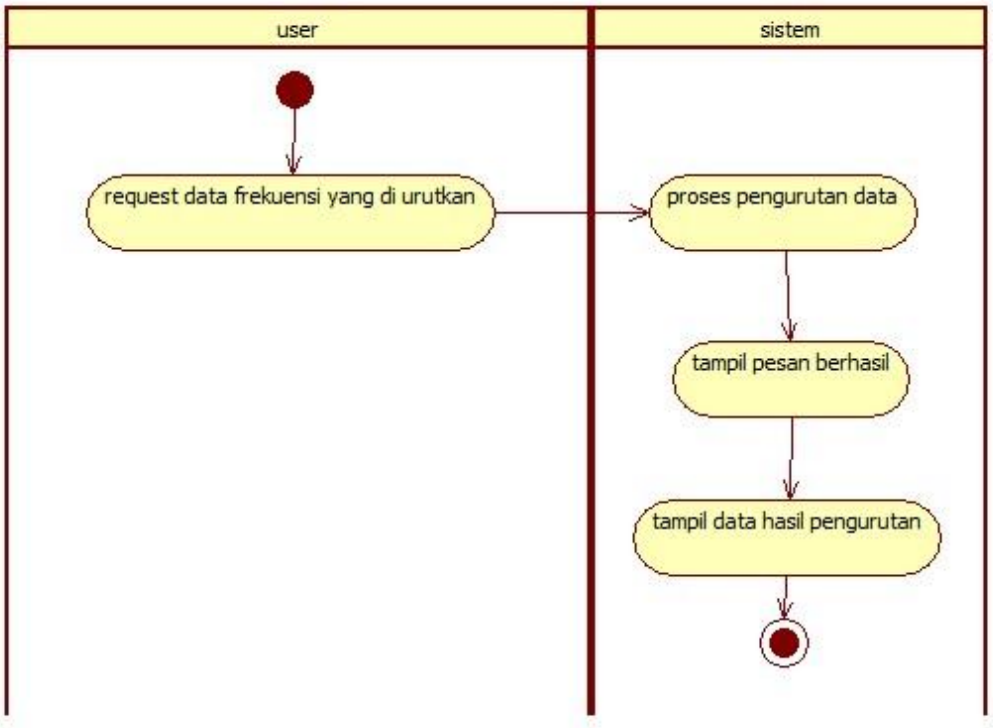
   Activity diagram Hitung Frekuensimenjelaskan aktivitas *user* dan sistem dalam melakukan proses hitung frekuensi. *Activity diagram* Hitung Frekuensi dapat dilihat pada gambar 3.6



Gambar 3. 36 Activity Diagram Hitung Frekuensi

6. *Activity Diagram* Pengurutan Data

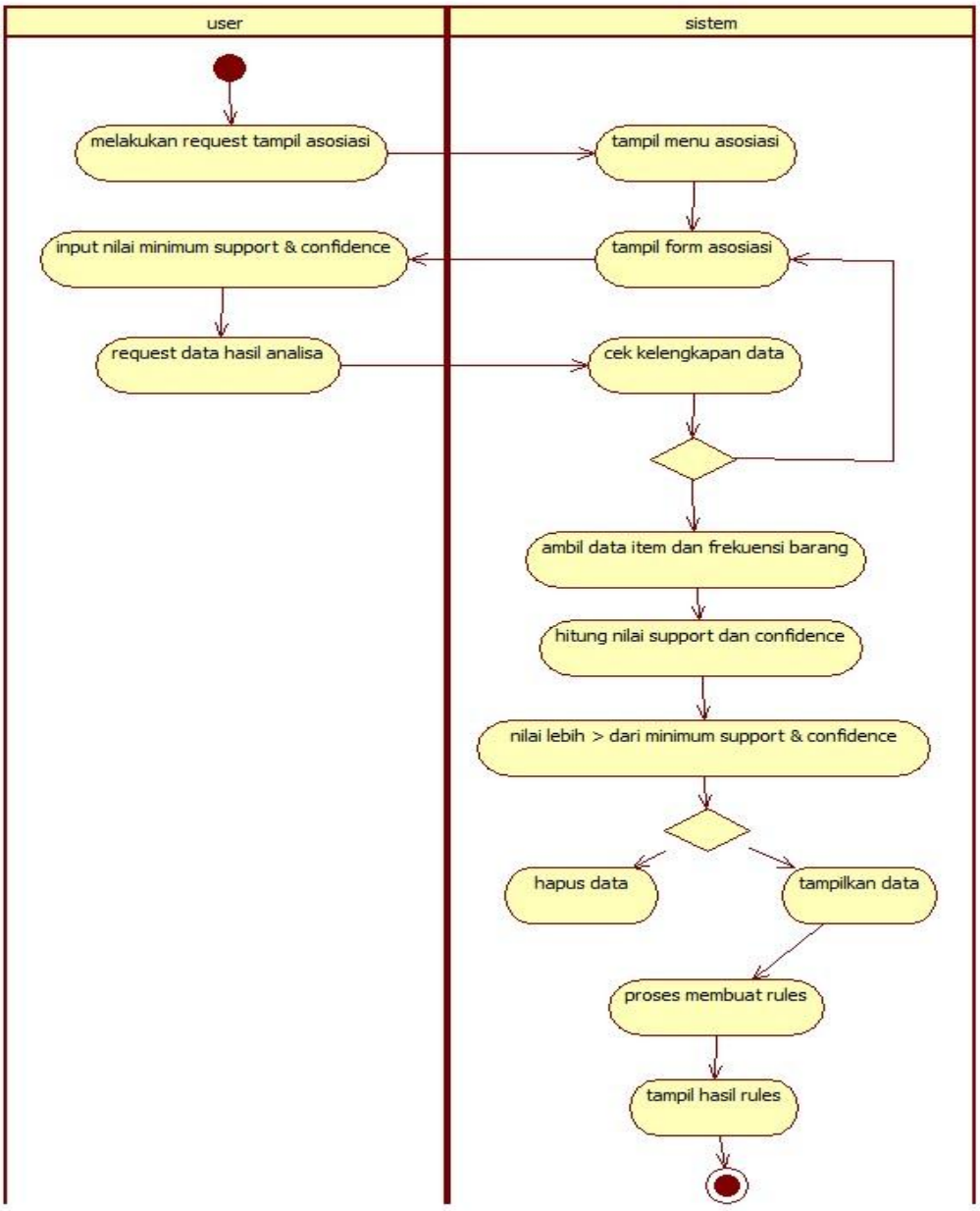
   Activity diagram Pengurutan Datamenjelaskan aktivitas *user* dan sistem dalam melakukan proses Pengurutan Data. *Activity diagram* Pengurutan Data dapat dilihat pada gambar 3.7



Gambar 3. 37 Activity Diagram Pengurutan Data

7. *Activity Diagram* Asosiasi

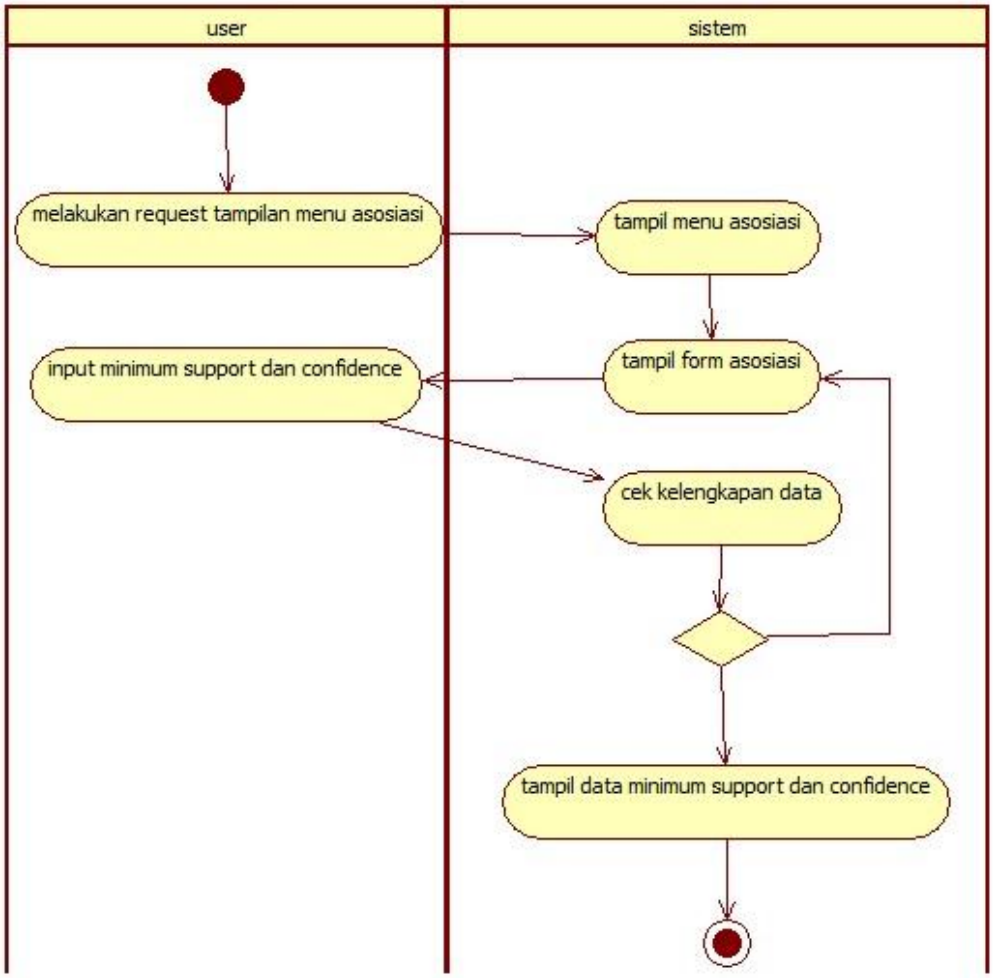
   Activity diagram asosiasimenjelaskan aktivitas *user* dan sistem dalam melakukan proses input parameter dan melihat rule. *Activity diagram* Asosiasi dapat dilihat pada gambar 3.8



Gambar 3. 38 Activity Diagram Asosiasi

8. *Activity Diagram* Input Parameter

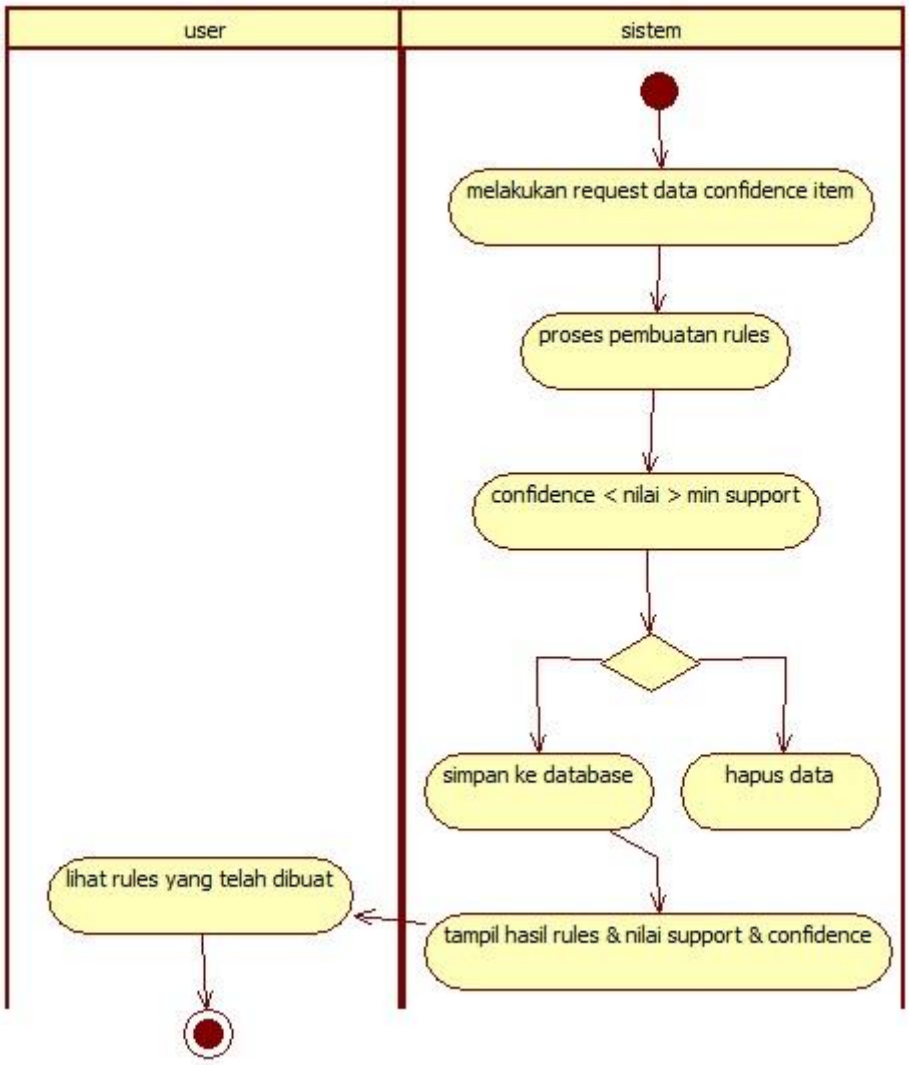
   Activity diagram input parametermenjelaskan aktivitas *user* dan sistem dalam melakukan proses memasukkan nilai *Minimum Support* dan *Minimum Confidence*. *Activity diagram* Input Parameterdapat dilihat pada gambar 3.9



Gambar 3. 39 Activity Diagram Input Parameter

9. *Activity DiagramRules*

   Activity diagram lihat hasil rule menjelaskan aktivitas *user* dan sistem dalam menampilkan hasil seleksi *rule* sesuai nilai *Minimumconfidence* yang dimasukkan oleh *user*. *Activity diagram* lihat hasil asosiasi dapat dilihat pada gambar 3.9

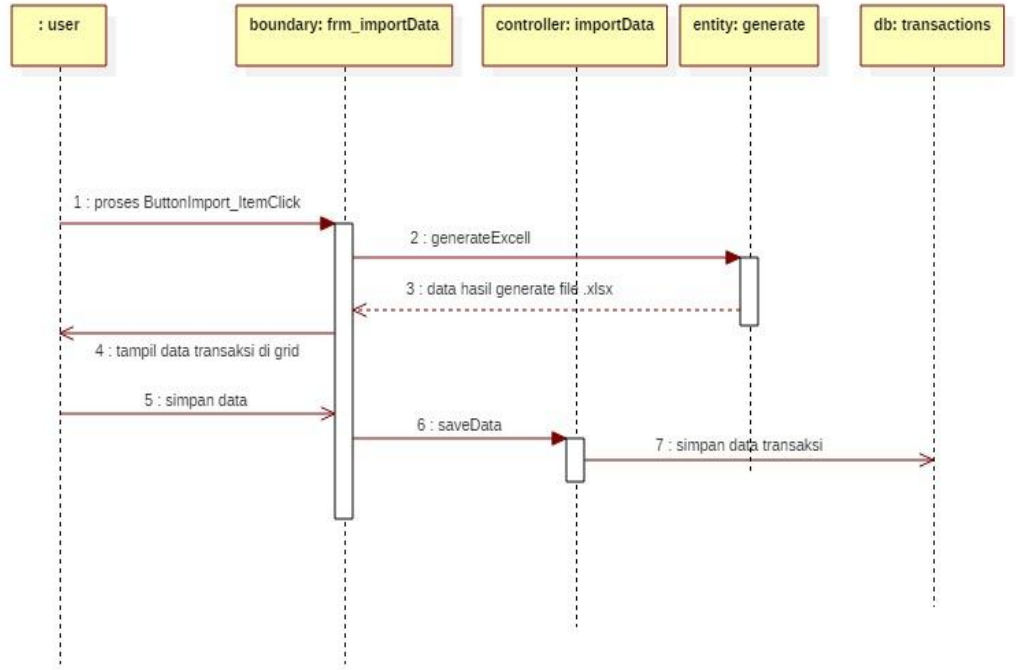


Gambar 3. 40 Activity Diagram Lihat Rule

3.2.3 Sequense Diagram

Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai tujuan hasil. Berikut adalah *sequence diagram* pada aplikasi *data mining* yang sedang dibangun pada Gambar 3. 10 sampai dengan Gambar 3.

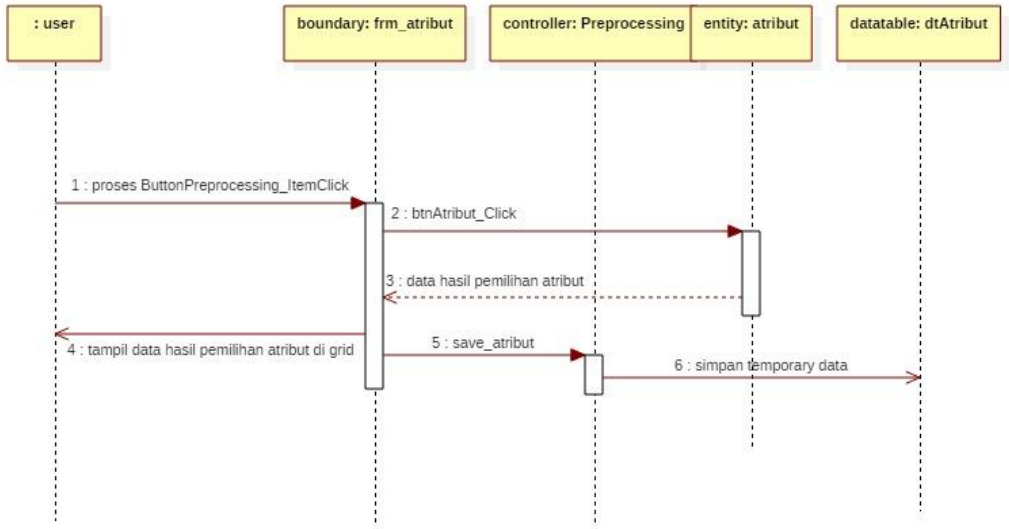
1. *Sequence Diagram Import Data*



Gambar 3. 41 Sequence Diagram Import Data

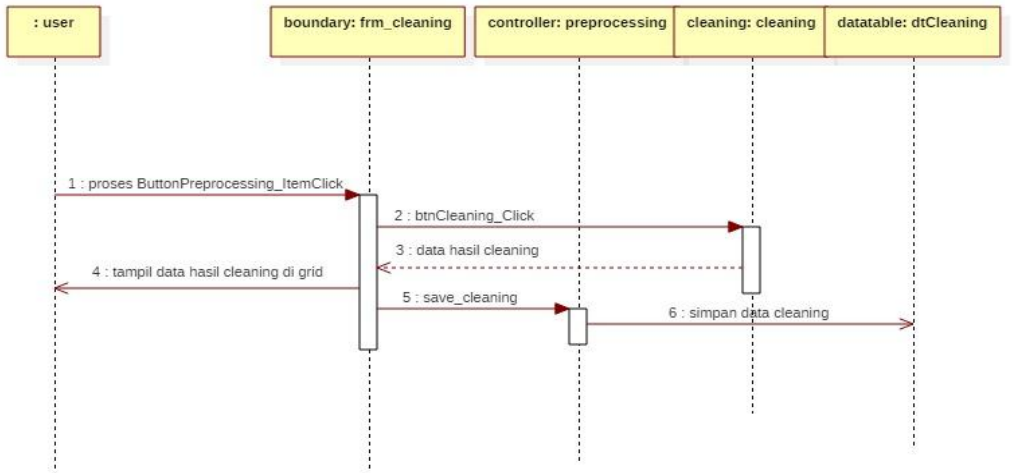
2. *Sequence Diagram Preprocessing*

   a. *Sequence Diagram Data Pemilihan Atribut*



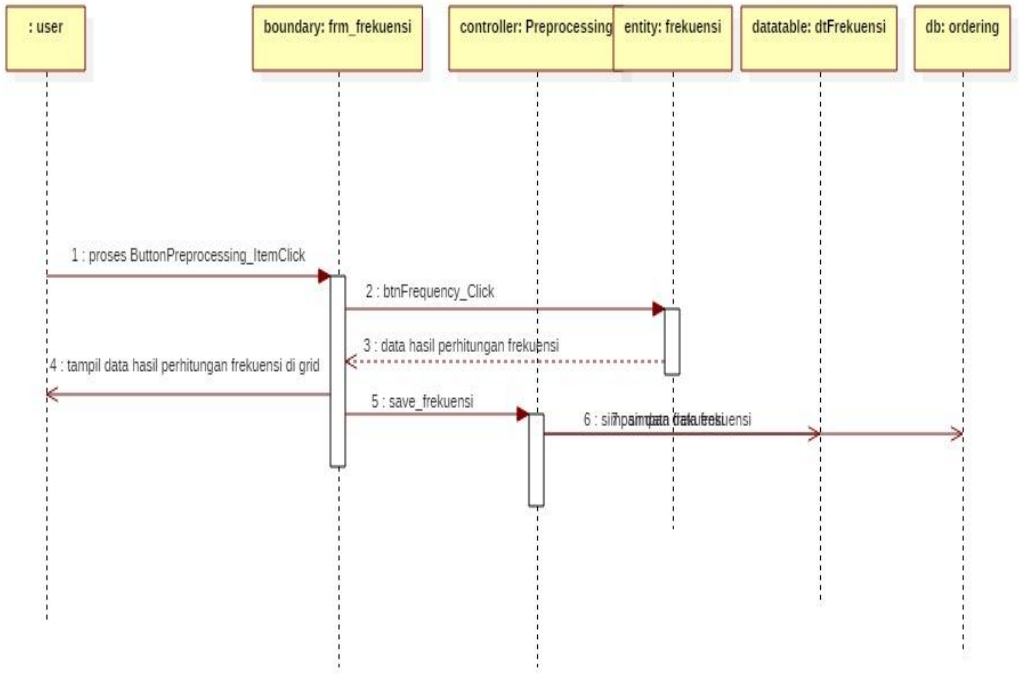
Gambar 3. 42 Sequence Diagram Data Pemilihan Atribut

b. Sequence Digram Data Pembersihan Data



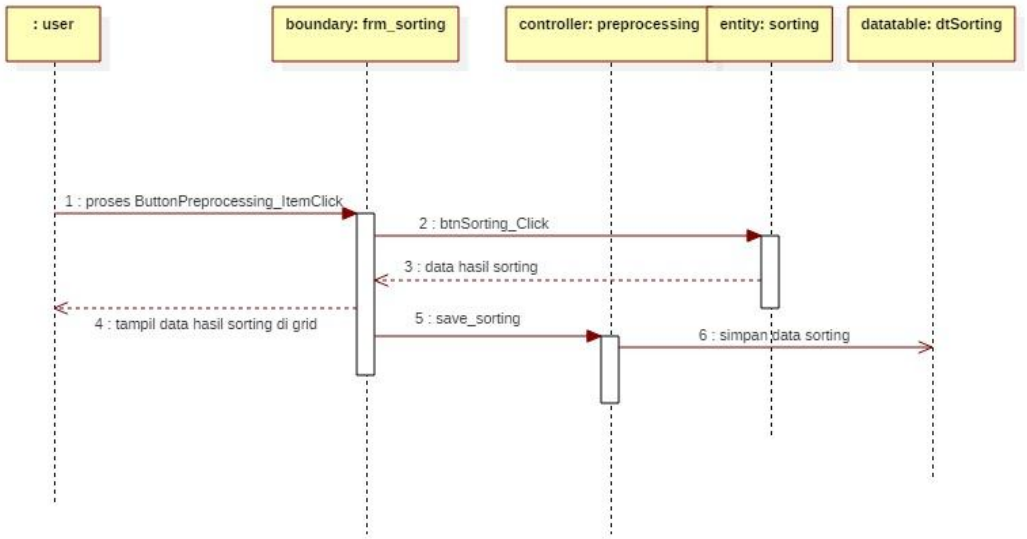
Gambar 3. 43 Sequence Digram Data Pembersihan Data

c. *Sequence Diagram* Hitung Frekuensi



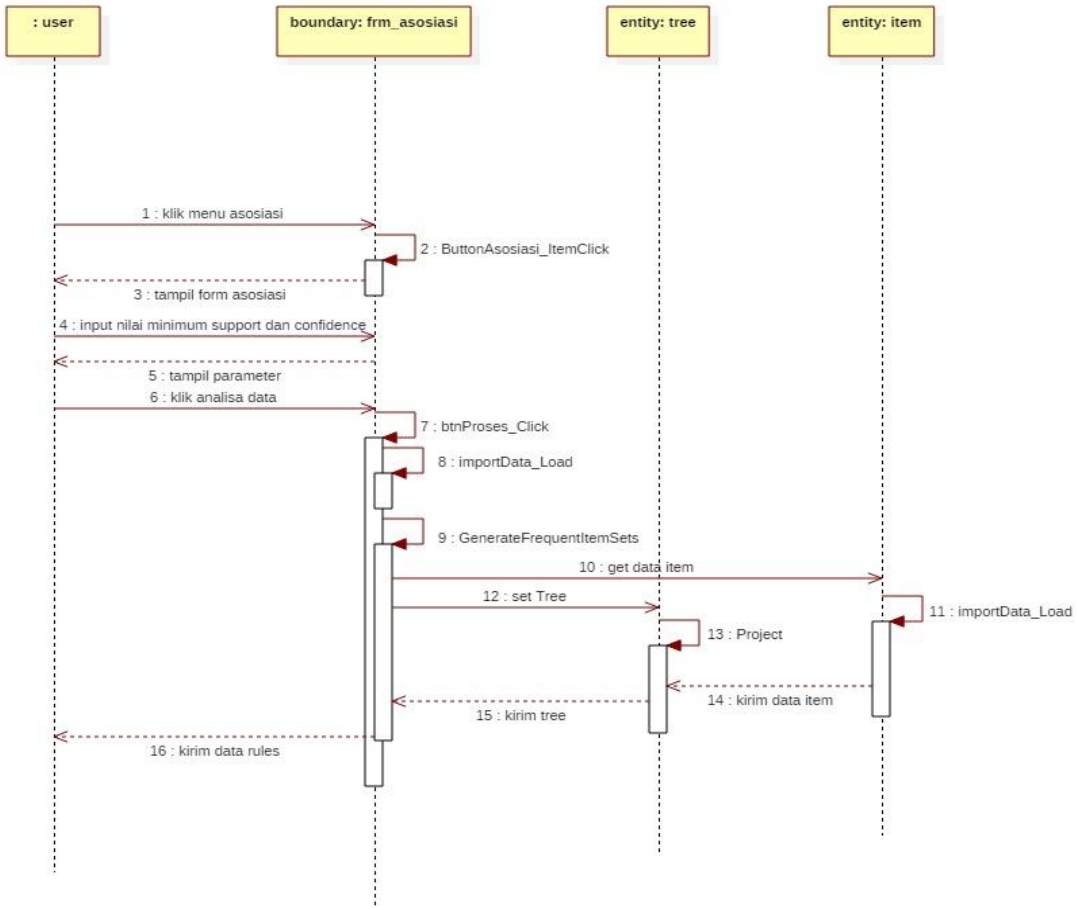
Gambar 3. 44 Sequence Digram Hitung Frekuensi

d. *Sequencediagram*Pengurutan Data



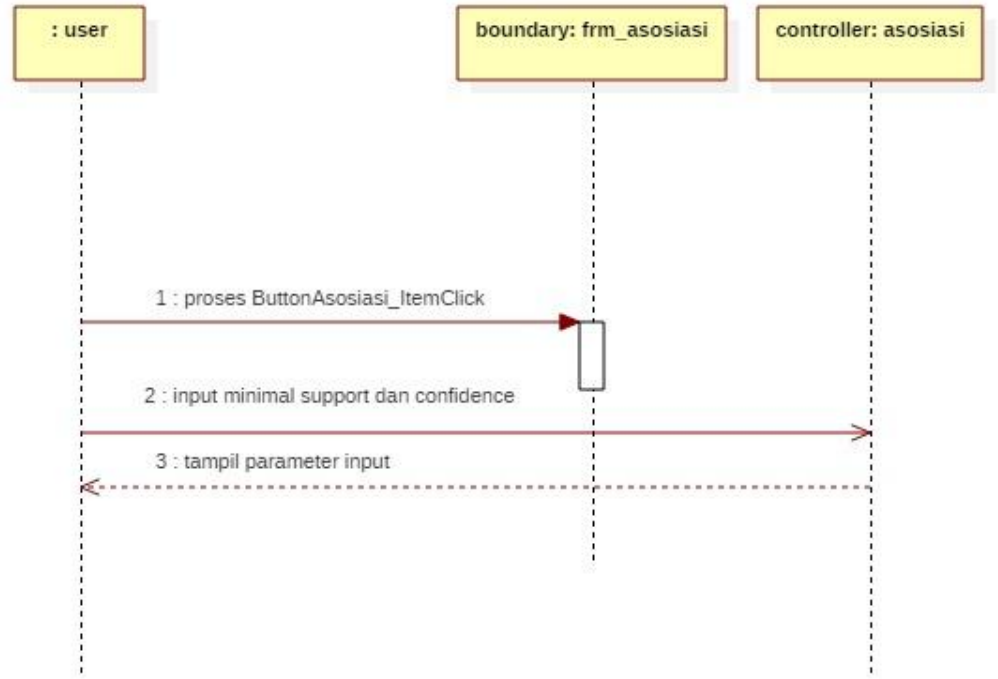
Gambar 3. 45 Sequense Diagram Pengurutan Data

e. *Sequence Diagram* Asosiasi



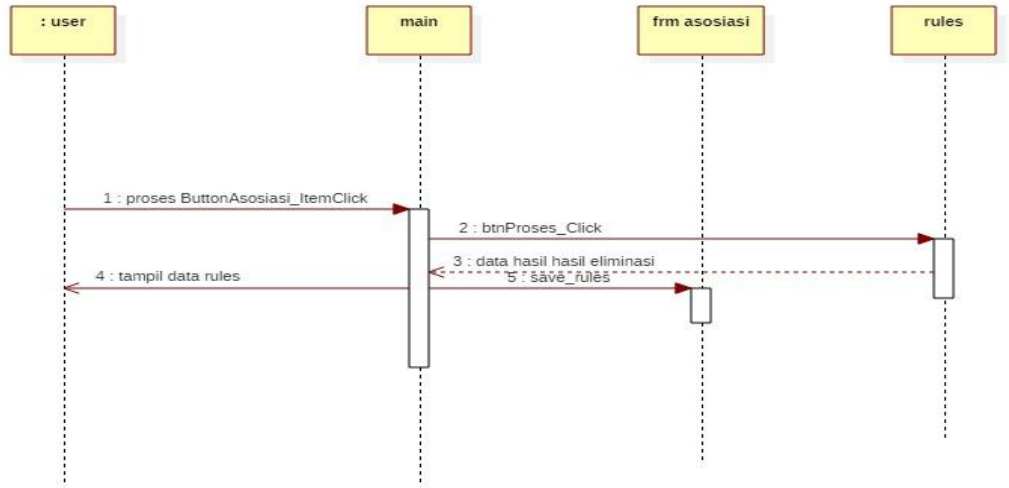
Gambar 3. 46 Sequense Diagram Asosiasi

f. *Sequence Diagram* Input Parameter



Gambar 3. 47 Sequense Diagram Input Parameter

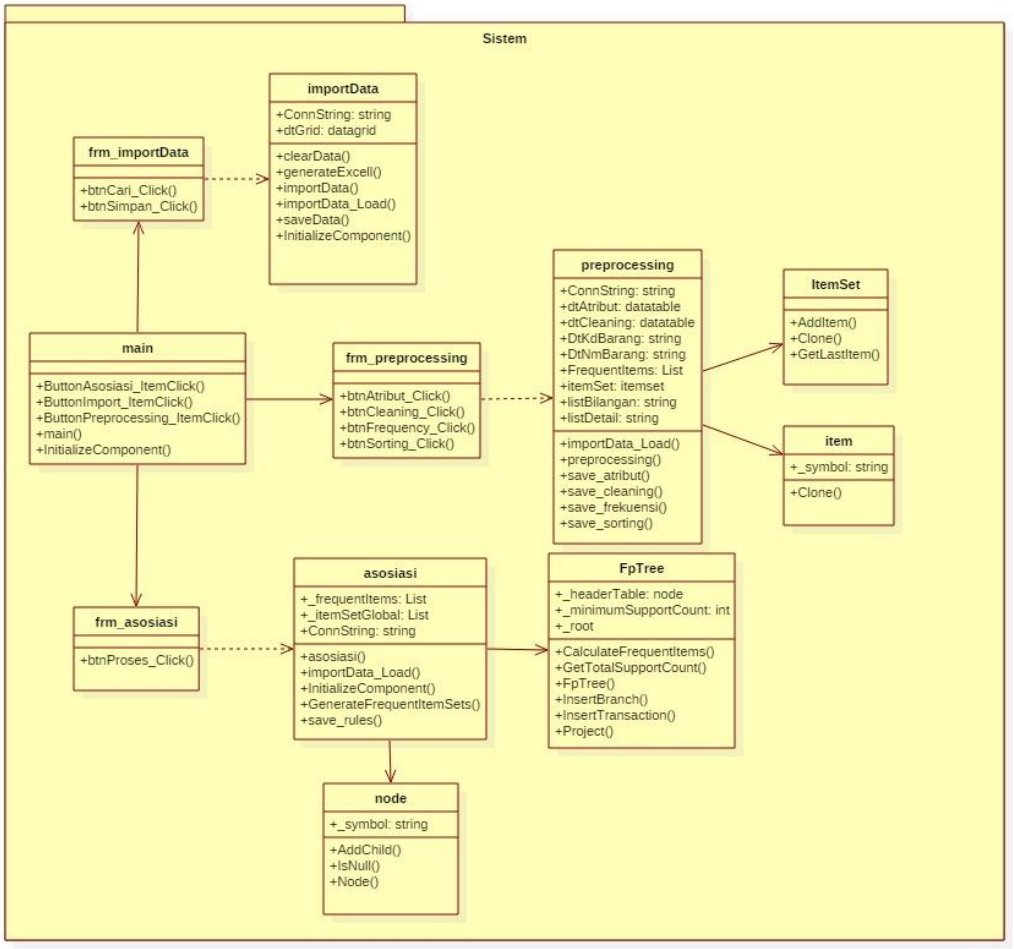
g. *Sequence DiagramRules*



Gambar 3. 48 Sequense Diagram Rules

3.2.4 Class Diagram

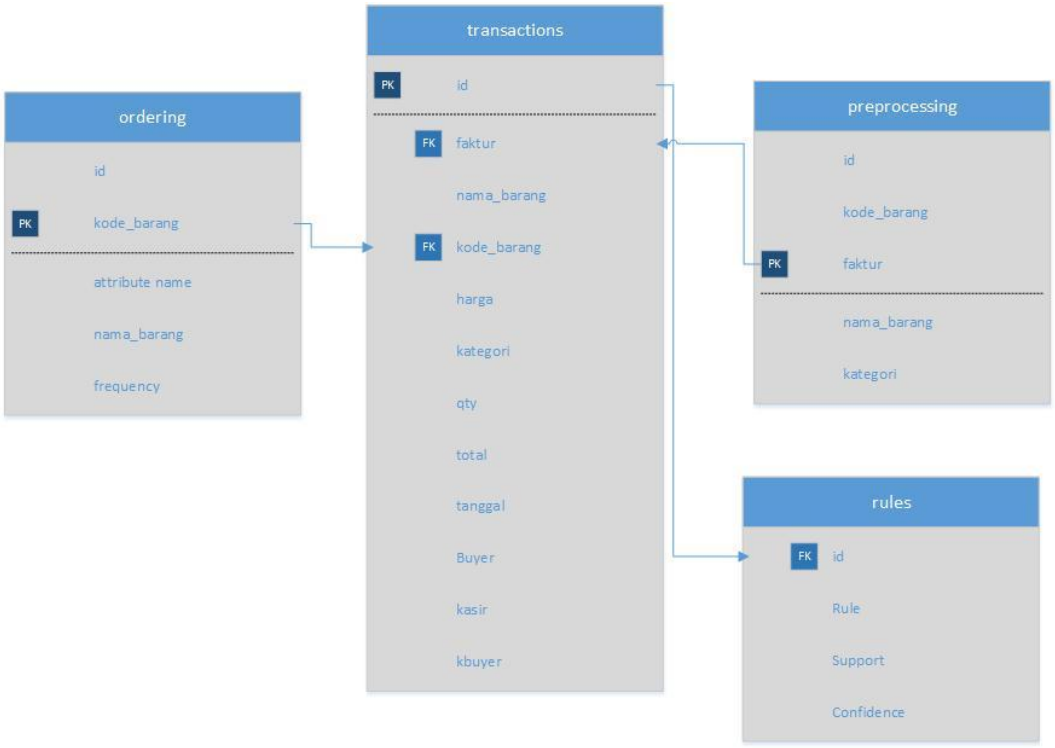
Berikut adalah *class diagram* dari sistem yang akan dibangun. Hasilnya dapat pada gambar 3.18



Gambar 3. 49 Class Diagram

3.2.5 Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Berikut ini adalah skema relasi pada sistem pembentukan paket produk di Minimarket Warga Tunggal Bandung :



Gambar 3. 50 Skema Relasi

3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional pada penelitian ini terdiri dari analisis spesifikasi kebutuhan perangkat keras / *hardware*, analisis kebutuhan perangkat lunak / *software*, dan analisis kebutuhan pengguna / *user*.

3.3.1 Analisis Kebutuhan Perangkat Keras

Analisis spesifikasi kebutuhan perangkat keras bertujuan untuk mengetahui spesifikasi perangkat keras yang digunakan. Berdasarkan pengamatan yang dilakukan di Minimarket Warga Tunggal Bandung, spesifikasi perangkat keras yang digunakan oleh Minimarket Warga Tunggal Bandung adalah sebagai berikut:

Tabel 3. 27Spesifikasi Perangkat Keras yang Sedang Berjalan

No	Perangkat Keras	Spesifikasi
1	RAM	1 GB
2	Harddisk	160 GB
3	Monitor	LCD 14''
4	Processor	Intel Core 2 Duo E7500

Adapun spesifikasi perangkat keras yang dibutuhkan untuk mengimplementasikan sistem yang akan dibangun adalah sebagai berikut:

Tabel 3. 28 Spesifikasi Perangkat Keras yang dibutuhkan

No	Perangkat Keras	Spesifikasi
1	RAM	512MB
2	Harddisk	160 GB
3	Monitor	LCD 14''
4	Processor	Intel Dual Core 2.93 GHz

Melihat spesifikasi perangkat keras yang digunakan di Minimarket Warga Tunggal Bandung seperti pada tabel 3.16 dan tabel spesifikasi perangkat keras yang dibutuhkan seperti pada tabel 3.17 maka dapat disimpulkan bahwa semua perangkat keras yang dibutuhkan telah memadai untuk menjalankan sistem.

3.3.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dalam pembangunan aplikasi digunakan untuk mengetahui dan mengoptimalkan implementasi dari aplikasi yang akan dibangun.

1. Analisis spesifikasi perangkat lunak pada sistem yang sedang berjalan di Minimarket Warga Tunggal Bandung adalah:
   a. Sistem Operasi Windows 7
   b. Microsoft Office 2013
2. Analisis spesifikasi kebutuhan perangkat lunak untuk menjalankan sistem yang akan dibangun adalah :

a. Sistem Operasi Windows 7
b. MySQL sebagai database server
c. Microsoft Visual Studio 2010

3. Evaluasi kebutuhan perangkat lunak

   Berdasarkan analisis spesifikasi perangkat lunak yang ada di Minimarket Warga Tunggal Bandung, spesifikasi perangkat lunak yang digunakan masih kurang. Untuk mengatasi hal tersebut akan di install MySQL dan juga Visual Studio pada komputer yang digunakan

3.3.3 Analisis Kebutuhan Pengguna

User atau pengguna yang akan mengoperasikan aplikasi ini merupakan manager yang bertindak sebagai pengolah bisnis. Analisis Pengguna dapat dilihat pada tabel 3.14

Tabel 3. 29 Analisis Pengguna

User	Akses	Keahlian
Manager	Manager dapat mengolah data transaksi penjualan dan memperoleh informasi dari aplikasi *mining*.	Dapat mengoperasikan aplikasi dengan baik seperti penginputan data transaksi, penghapusan data transaksi, dan juga dapat mengolah informasi yang didapat.

3.4 Perancangan Sistem

3.4.1 Perancangan Kelas

Berikut adalah perancangan kelas yang ada di Class Diagram :

Tabel 3. 30Perancangan class main

Class : main	
Nama	Type
ButtonAsosiasi_ItemClick()	Public
ButtonImport_ItemClick()	Public
ButtonPreprocessing_ItemClick()	Public
Main()	Public
InitializeComponent()	

Tabel 3. 31Perancangan Class frm_importData

Class : frm_importData	
Nama	Type
btnCari_Click()	Public
btnSimpan_Click()	Public

Tabel 3. 32Perancangan Class ImportData

Class : importData	
Nama	Type
ConnString: string	Public
dtGrid: datagrid	Public
clearData()	Public
generateExcell()	Public
importData()	Public
importData_Load()	Public
saveData()	Public
InitializeComponent()	Public

Tabel 3. 33Perancangan Class frm_prepocessing

Class : frm_preprocessing	
Nama	Type
btnAtribut_Click()	Public
btnCleaning_Click()	Public
btnFrequency_Click()	Public

Tabel 3. 34Perancangan Class prepocessing

Class : preprocessing	
Nama	Type
ConnString: string	Public
dtAtribut: datatable	Public
dtCleaning: datatable	Public
DtNoFaktur: string	Public
DtNmBarang: string	Public
FrequentItems: List	Public
itemSet: itemset	Public
listBilangan: string	Public
listDetail: string	Public
importData_Load()	Public
preprocessing()	Public
save_atribut()	Public
save_cleaning()	Public
save_frekuensi()	Public
save_sorting()	Public

Tabel 3. 35Perancangan Class ItemSet

Class : ItemSet	
Nama	Type
AddItem()	Public
Clone()	Public
GetLastItem()	Public

Tabel 3. 36Perancangan Class Item

Class : Item	
Nama	Type
_symbol: string	Public
Clone()	Public

Tabel 3. 37Perancangan Class frm_asosiasi

Class : frm_asosiasi	
Nama	Type
btnProses_Click()	Public

Tabel 3. 38Perancangan Class asosiasi

Class : asosiasi	
Nama	Type
_frequentItems: List	Public
_itemSetGlobal: List	Public
ConnString: string	Public
asosiasi()	Public
importData_Load()	Public
InitializeComponent()	Public
GenerateFrequentItemSets()	Public
save_rules()	Public

Tabel 3. 39Perancangan Class node

Class : node	
Nama	Type
_symbol: string	Public
AddChild()	Public
IsNull()	Public
Node()	Public

Tabel 3. 40Perancangan Class Fp_Tree

Class : FpTree	
Nama	Type
_headerTable: node	Public
_minimumSupportCount: int	Public
_root	Public
CalculateFrequentItems()	Public
GetTotalSupportCount()	Public
FpTree()	Public
InsertBranch()	Public
InsertTransaction()	Public
Project()	Public

3.4.2 Perancangan Data

3.4.2.1 Perancangan Basis Data

Perancangan Basis Data adalah menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama-sama. Tabel yang akan digunakan dalam program ini antara lain:

1. Tabel Transaction
2. Tabel Ordering
3. Tabel Preprocessing
4. Tabel Rules

3.4.2.2 Struktur Tabel

Berikut ini adalah struktur dari tabel-tabel yang digunakan pada sistem yang akan dibangun:

Tabel 3. 41 Tabel Yang Digunakan

No	Nama Tabel	Nama Atribut
1	<ordering>	<id, kode_barang,nama_barang, frequency>
2	<preprocessing>	<id, kode_barang, faktur, nama_barang, kategori>
3	<rules>	<Rule, Support, Confidence>
4	<transactions>	<id, faktur, nama_barang, kode_barang, harga, qty, total, tanggal, Buyer, kasir, kbuyer>

1. Tabel ordering

   Tabel ordering digunakan untuk menyimpan data barang yang dibeli.

Tabel 3. 42 Tabel Ordering

Nama Field	Tipe Data	Panjang	Keterangan
Id	int	11	
kode_barang	varchar	10	Primary key
nama_barang	varchar	100	
Frequency	int	11	

2. Tabel Preprocessing

   Tabel Preprocessing digunakan untuk menyimpan data hasil pemilihan atribut dan pembersihan data.

Tabel 3. 43 Tabel Preprocessing

Nama Field	Tipe Data	Panjang	Keterangan
id	Int	10	
kode_barang	varchar	10	
faktur	varchar	50	FOREIGN KEY (faktur) REFERENCES `transaction`(faktur)
nama_barang	varchar	100	
kategori	varchar	20	

3.Tabel Rules

Tabel rules digunakan untuk menyimpan hasil rule

Tabel 3. 44 Tabel Rules

Nama Field	Tipe Data	Keterangan
Id	Int	FOREIGN KEY (id) REFERENCES 'transactions'(id)
Rule	text	
Support	float	
Confidence	float	

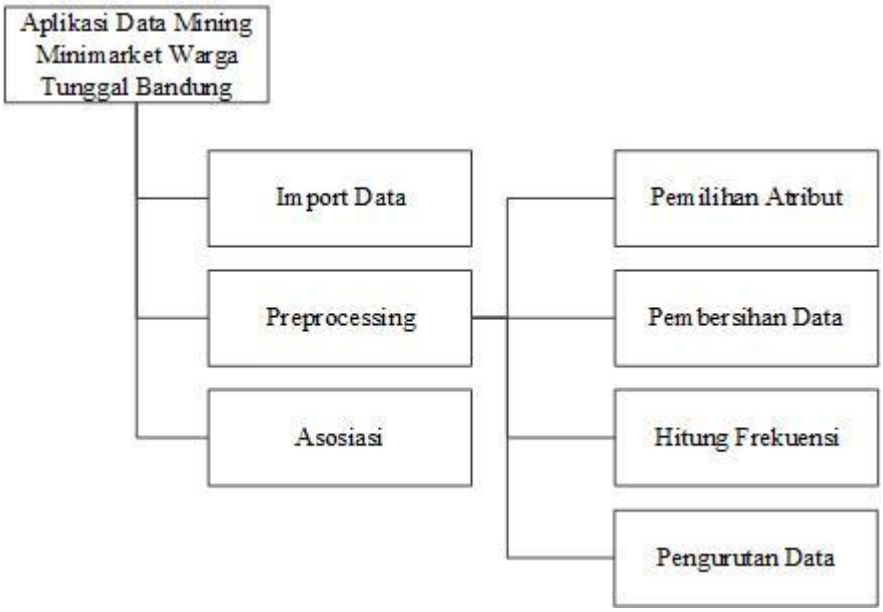
4.Tabel transactions

Tabel transactions adalah tabel yang digunakan untuk menyimpan data transaksi.

Tabel 3. 45 Tabel transactions

Nama Field	Tipe Data	Panjang	Keterangan
id	Int	10	
faktur	varchar	50	
nama_barang	text		
kode_barang	text		FOREIGN KEY (kode_barang) REFERENCES `ordering`(kode_barang)
harga	Int	10	
kategori	text		
qty	Int	10	
total	Int	10	
tanggal	text		
Buyer	varchar	10	
kasir	varchar	10	
kbuyer	varchar	10	

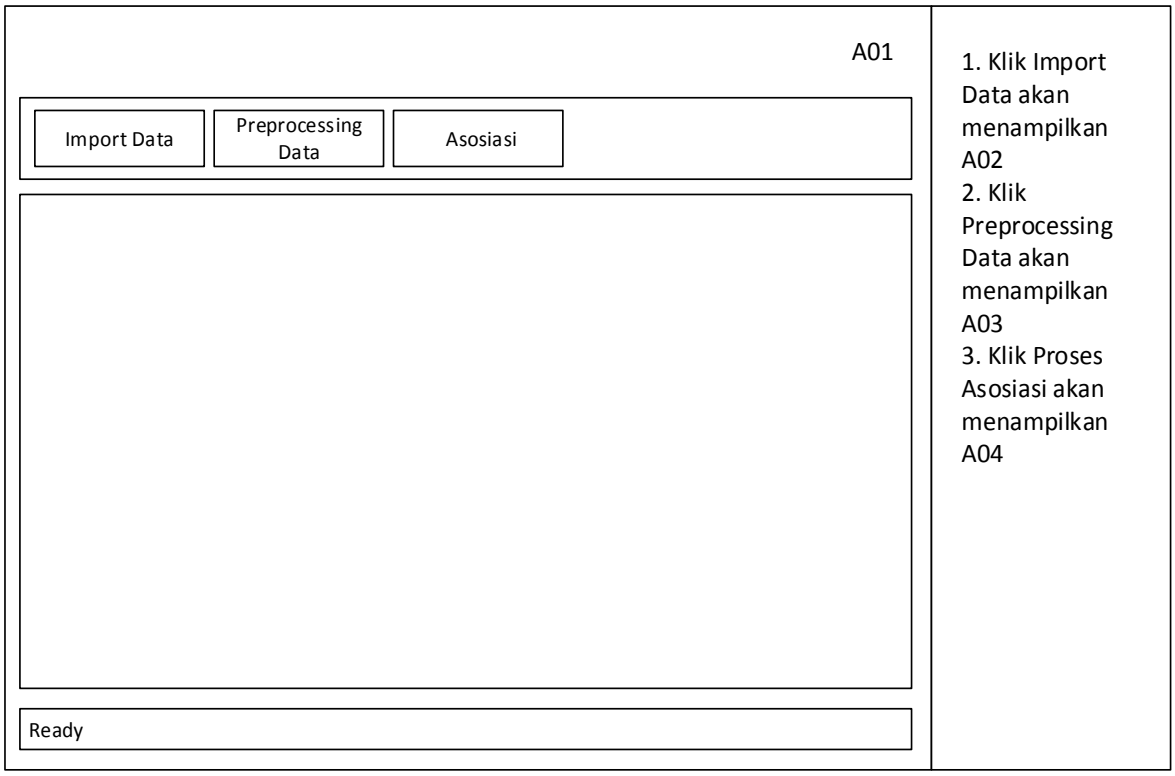
1.4.3 Perancangan Struktur Menu



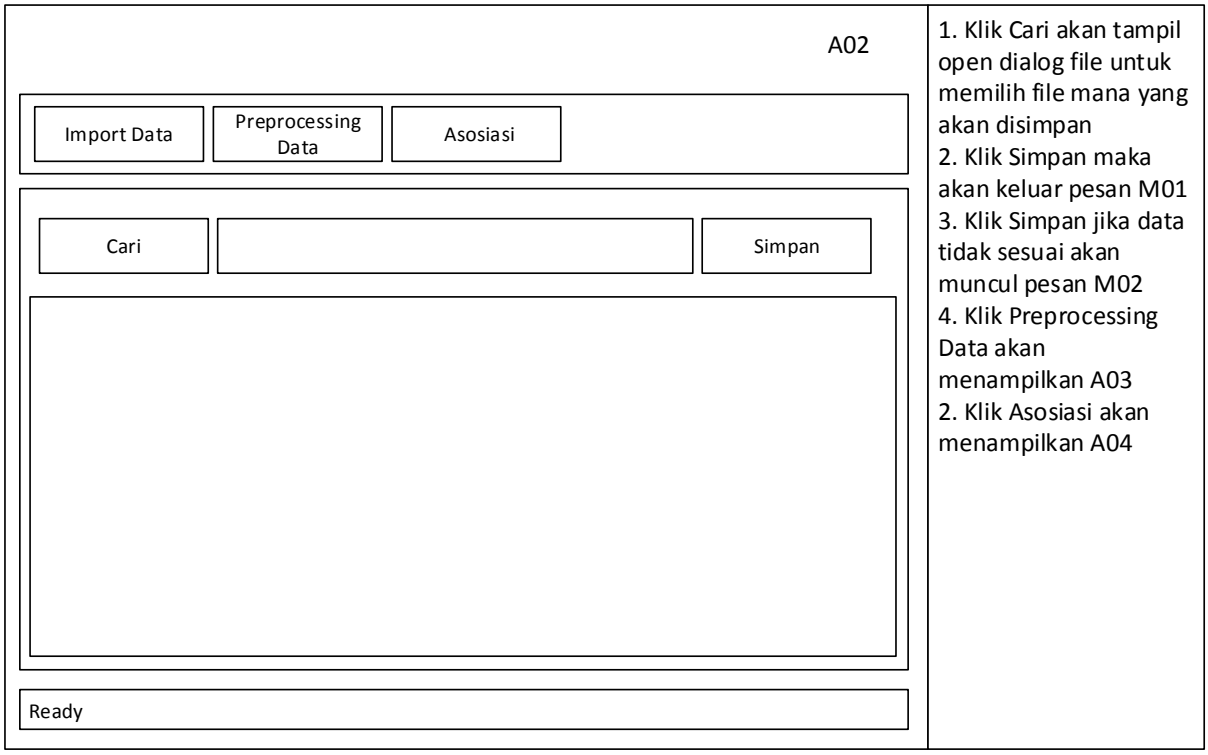
Gambar 3. 51 Struktur Menu

3.4.4 Perancangan Antar Muka

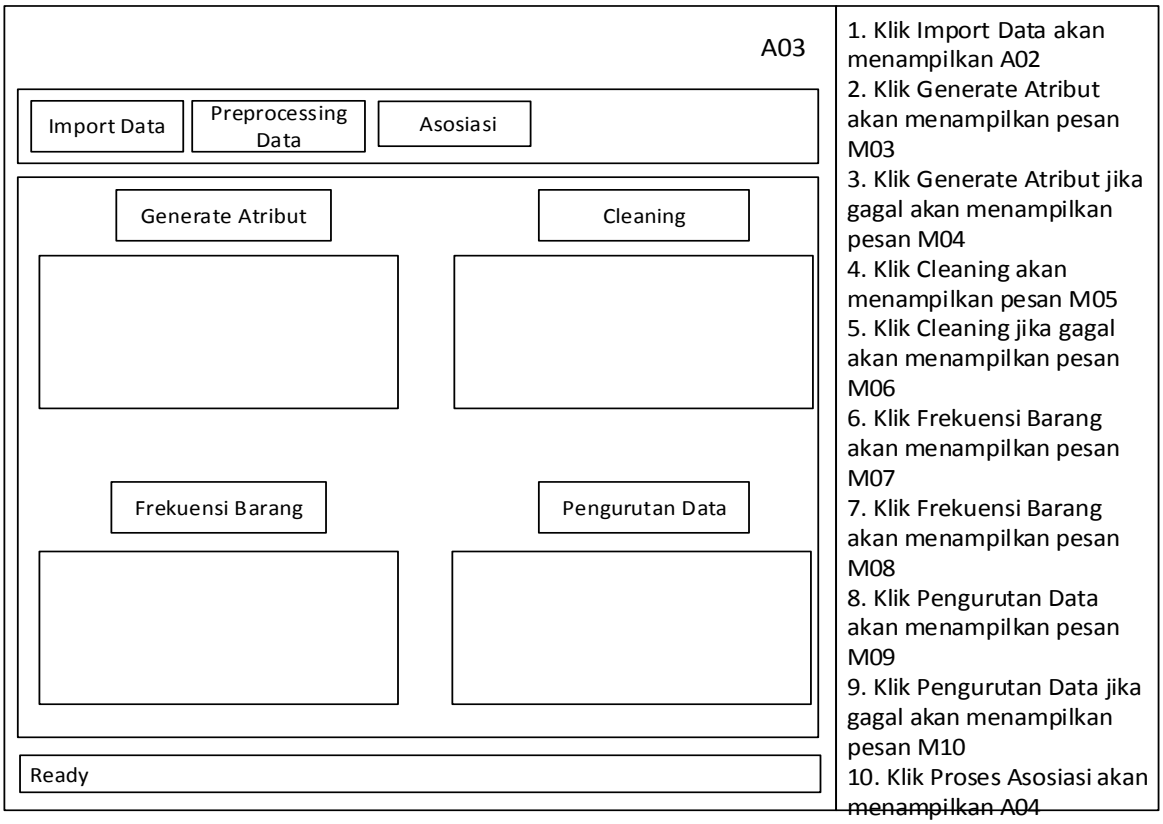
Perancangan antar muka berfungsi untuk memberi gambaran tentang pernagkat lunak yang akan dibangun, supaya lebih memudahkan dalam pembuatan perangkat lunak nantinya. Adapun perancangan antar muka untuk perangakt lunak yang akan dibangun akan dijelaskan berikut ini :



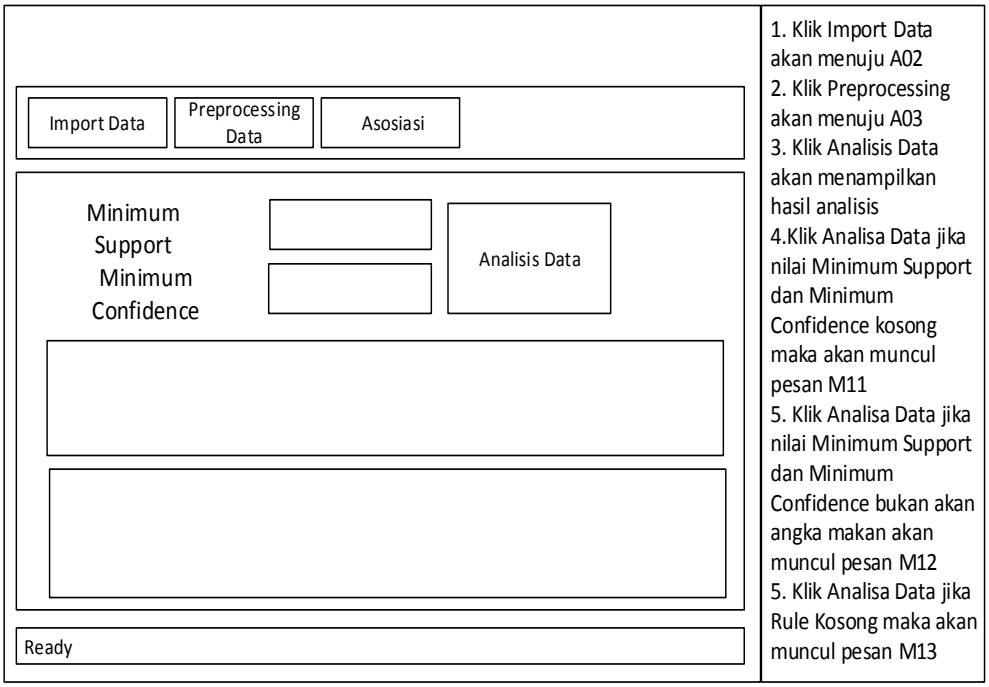
Gambar 3. 52 Perancangan Antar Muka Menu Utama



Gambar 3. 53 Perancangan Antar Muka Import Data



Gambar 3. 54Perancangan Antar Muka Preprocessing



Gambar 3. 55 Perancangan Antar Muka Asosiasi

3.4.5 Perancangan Pesan

Adapun perancangan pesan dapat dilihat pada gambar dibawah ini :

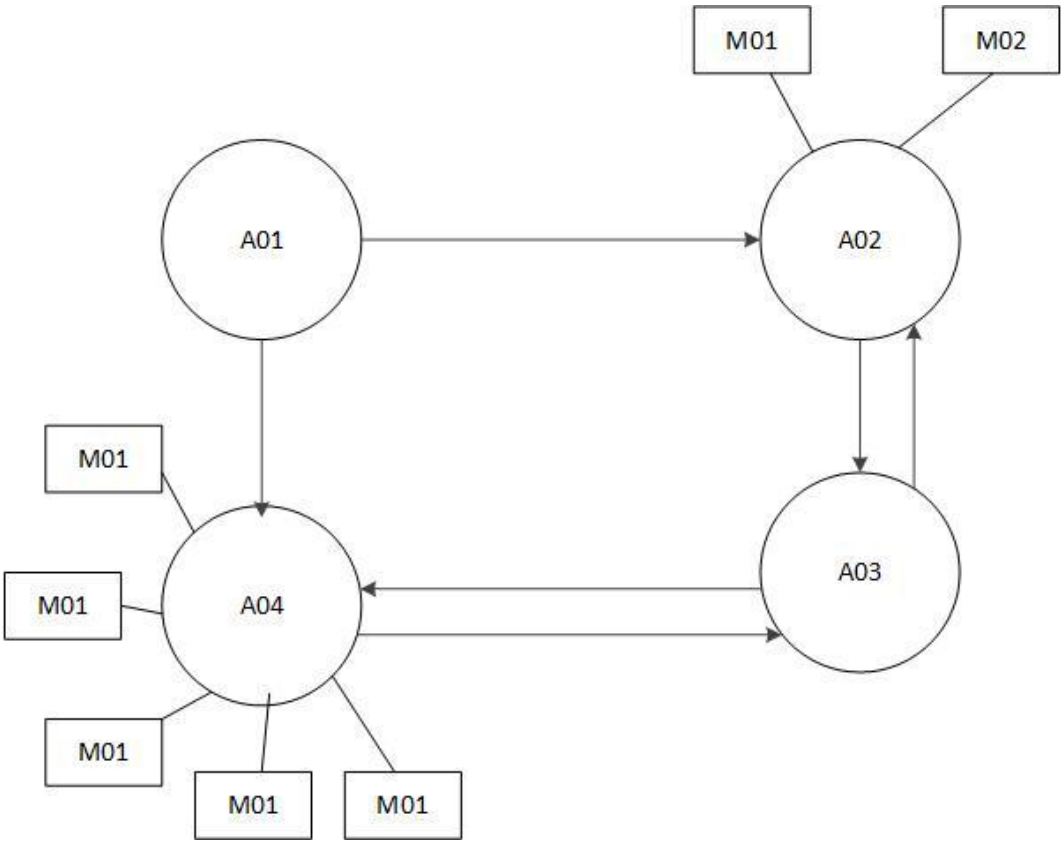
Tabel 3. 46 Perancangan Pesan

Kode Pesan	Pesan	No. Form
M01	Data Berhasil Disimpan	A02
M02	Data Gagal Disimpan	A02
M03	Selected Atribut Berhasil	A03
M04	Selected Atribut Gagal	A03
M05	Cleaning Data Berhasil	A03
M06	Cleaning Data Gagal	A03
M07	Hitung Frekuensi Kemunculan Item Berhasil	A03
M08	Hitung Frekuensi Kemunculan Item Gagal	A03
M09	Sorting Data Berhasil	A03

M10	Sorting Data Gagal	A03
M11	Minimum Support dan Minimum Confidence tidak boleh kosong	A04
M12	Error Proses Analisa	A04
M13	Rules tidak ditemukan	A04

3.4.6 Jaringan Semantik

Jaringan semantik bertujuan untuk merepresentasikan pemetaan data untuk mencegak duplikasi data. Adapun jaringan semantik yang menghubungkan antarmuka pada aplikasi yang akan dibangun dapat dilihat pada gambar berikut ini:



Gambar 3. 56 Jaringan Semantik