BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem didefinisikan sebagai penguraian suatu sistem informasi yang utuh kedalam beberapa bagian dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan ataupun hambatan-hambatan yang terjadi untuk menghasilkan sebuah solusi guna melakukan perbaikan-perbaikan terhadap sistem yang sedang berjalan. Dalam analisis sistem ini meliputi beberapa bagian, yaitu:

1. Analisis Masalah
2. Analisis Penerepan *Data Mining*
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. Perlu ada analisis masalah melalui rumusan masalah yang sudah ditentukan sebelumnya. Berdasarkan hasil pengamatan, dapat disimpulkan bahwa permasalahan yang ada di Primer Koperasi Kartika yaitu belum adanya informasi penentuan paket produk yang sesuai dengan minat konsumen.

3.1.2 Analisis Penerapan *Data Mining*

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 Primer Koperasi Kartika mempunyai tujuan bisnis yaitu untuk memenuhi permintaan pelanggan sehingga dapat meningkatkan kepuasan anggota Bekangdam III / Siliwangi ataupun pelanggan umum.

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.

3.1.2.2 Pemahaman Data

Setelah melakukan tahap pemahaman bisnis selanjutnya akan ada tahap pemahaman Data. Untuk memahami data, ada beberapa tahapan yang harus dilakukan, diantaranya sebagai berikut:

1. Mengumpulkan data awal

   Data dalam penelitian adalah *sample* data penjulan di Primer Koperasi Kartika yang berasal dari *Microsoft excel (*.xlsx) yang nantinya akan dilakukan proses *mining* pada program.

2. Menjelaskan data

   Berikut ini adalah penjelasan masing-masing atribut yang terdapat pada tabel transaksi penjualan berdasarkan Lampiran D pada tabel D-1 yang menampung 90 record data, dapat dilihat pada tabel 3.1:

Tabel 3. 1 Struktur Data Transaksi

Keterangan		
Fungsi	Untuk mengetahui produk apa saja yang sering dibeli oleh pembeli	
Format	*Microsoft Excel* (.xlsx)	
Atribut	Faktur	Nomor urut penjualan
	Kode_barang	Kode dari setiap barang
	Nama_barang	Nama barangyang dibeli
	Harga_satuan	Harga satuan masing-masing barang
	Qty	Jumlah barang yag dibeli
	Harga	Jumlah keseluruhan harga barang

3.1.2.3 Persiapan Data

Persiapan data adalah hal yang harus dilakukan, karena tidak semua atribut digunakan dalam proses data *mining.* Proses ini dilakukan agar data yang akan digunakan sesuai dengan kebutuhan. Adapun tahapan tahapan *preprocessing* adalah sebagai berikut:

1. Ekstrasi data

   Ekstrasi data adalah proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data. Dalam penelitian ini data diambil dari file *Microsoft excel.* Data yang digunakan adalah data transaksi penjualan sebanyak 90 record data terdapat 30 transaksi. Data transaksi penjualan terdapat di lampiran D tabel D-1.

2. Pemilihan atribut

   Pemilihan atribut adalah proses pemilihan atribut data yang digunakan, sehingga data tersebut dapat diproses sesuai dengan kebutuhan *data mining.* Dalam tahap ini atribut yang digunakan adalah Faktur dan Kode_Barang. Faktur merupakan identitas dari setiap transaksi, sedangkan Kode_Barang digunakan untuk mengetahui jenis barang apa saja yang dibeli oleh pelanggan. Hasil pemilihan Atribut dapat dilihat pada lampiran D tabel D-2.

3. Pembersihan data

Proses menghilangkan *noise* dan menghilangkan data yang tidak relevan disebut pembersihan data. Dalam penelitian ini data yang dihilangkan yaitu data penjualan yang hanya terdapat 1 barang yang terjual dalam 1 transaksi yang sama. Hasil pembersihan data dapat dilihat pada pada lampiran D tabel D-3.

3.1.2.4 Pemodelan

Pemodelan merupakan tahap untuk membuat model desain sistem yang akan dibangun. Data yang digunakan untuk pengolahan data ini adalah data terakhir yang sudah dilakukan proses pembersihan data pada Lampiran D tabel D-3. Dalam tahap ini akan dilakukan penyelesaian masalah pada perekomendasian paket produk di Primer Koperasi Kartika dengan metode Association Rule dengan menggunakan Algoritma FP-Growth. Tahap ini dilakukan dengan menggunakan metode *association rule* yang akan memasuki ke dalam dua tahap yaitu:

1. Analisis pola frekuensi tinggi.
2. Pembentukan aturan asosiasi.

Pada tahap awal dalam pemodelan adalah menentukan *minimum support*. *Minimum support* adalah suatu ukuran yang menunjukan seberapa besar tingkat suatu kombinasi suatu barang pada sebuah data transaksi. Untuk kasus nilai *minimum support* yang digunakan yaitu 10% dari 30 data transaksi penjualan yang digunakan yaitu sebesar 3, yang berarti barang yang memiliki jumlah *frequent* kurang dari 3 akan dieliminasi dalam data transaksi. Langkah pertama yang dilakukan untuk menghitung nilai *support* dari sebuah barang adalah dengan menghitung kemunculan tiap barang yang sudah disusun secara *ascending.* Jumlah kemunculan setiap barang dapat dilihat pada Tabel 3. 2 Frequent Item:

Tabel 3. 2 Frequent Item

Kode_barang	Support Count
11	15
201	7
19	11

Kode_barang	Support Count
18	11
210	6
12	5
20	4
10	4
212	4
40	2
60	3
146	3
171	2
45	2
58	2
214	2
2	1
9	1
13	1
27	1
44	1
57	1
133	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 *support* dapat dilihat pada Tabel 3. 3 Item yang memenuhi Minimum Support:

Tabel 3. 3 Item yang memenuhi Minimum Support

Kode_barang	Support Count
11	15
19	11
18	11
201	7
210	6
12	5
10	4
20	4
212	4
60	3

Kode_barang	Support Count
146	3

Setelah didapatkan data yang memenuhi nilai *minimum support* seperti pada tabel 3.3, selanjutnya mencari *Frequency of Occurence*, urutkan tabel 3.3 berdasarkan *frequent* terbesar sampai dengan terkecil, kemudian berikan nilai *priority* dimulai dari data *frequent* terbesar sampai dengan terkecil. Jika ada Kode_Barang dengan jumlah kemunculan yang sama maka yang di dahulukan adalah Kode_Barang yang pertama kali muncul dalam transaksi. Hasil pemberian nilai priority dapat dilihat pada Tabel 3. 4

Tabel 3. 4 Pemberian nilai priority

Kode_barang	Support Count	Priority
11	15	1
19	11	2
18	11	3
201	7	4
210	6	5
12	5	6
10	4	7
20	4	8
212	4	9
60	3	10
146	3	11

Setelah semua *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 akan didahulukan penulisannya.

Hasil pengurutan berdasarkan *priority* dapat dilihat pada Tabel 3. 5 Hasil pengurutan berdasarkan *priority* berikut ini :

Tabel 3. 5 Hasil pengurutan berdasarkan *priority*

Faktur	Kode_Barang
1101	11
1101	19
1101	18
1102	11
1102	201
1102	12
1102	10
1103	11
1103	18
1103	12
1104	19
1104	201
1105	19
1105	210
1105	12
1105	10
1106	11
1106	18
1107	210
1107	12
1107	10
1108	11
1108	20
1109	18
1109	10
1109	20
1110	11
1110	19
1110	18
1110	12
1111	19
1111	18
1112	11
1112	18
1113	19
1113	201
1113	212
1114	11
1114	201

Faktur	Kode_Barang
1115	19
1115	210
1117	11
1117	210
1118	20
1118	212
1119	11
1119	201
1121	11
1121	19
1121	18
1121	60
1122	18
1122	20
1123	19
1123	212
1123	60
1123	146
1124	11
1124	201
1126	11
1126	19
1126	18
1127	201
1127	210
1127	212
1128	11
1128	146
1129	19
1129	18
1129	210
1129	146

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 D pada tabel D-4.

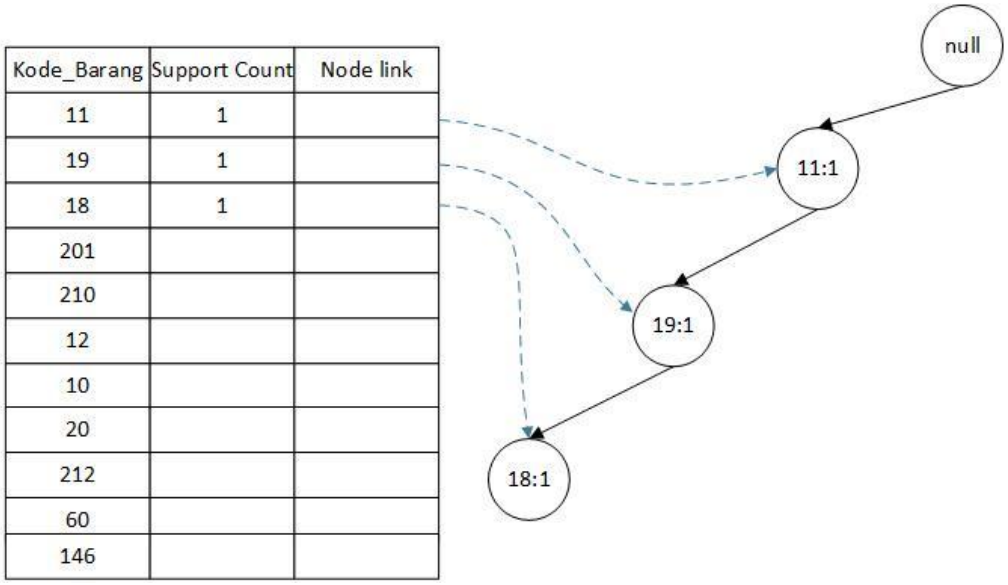
1. Setelah tahap pembangunan *fp-tree* dari sekumpulan data transaksi dilakukan, maka akan dilakukan proses untuk mencari *frequent itemset* yang signifikan. Berikut adalah tahapanya :
   a) Tahap Pembangkitan *Conditional Pattern Base*

      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 f*requent itemset* apabila c*onditional 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 1101 dengan produk yang dipesan adalah 11,19, 18 :

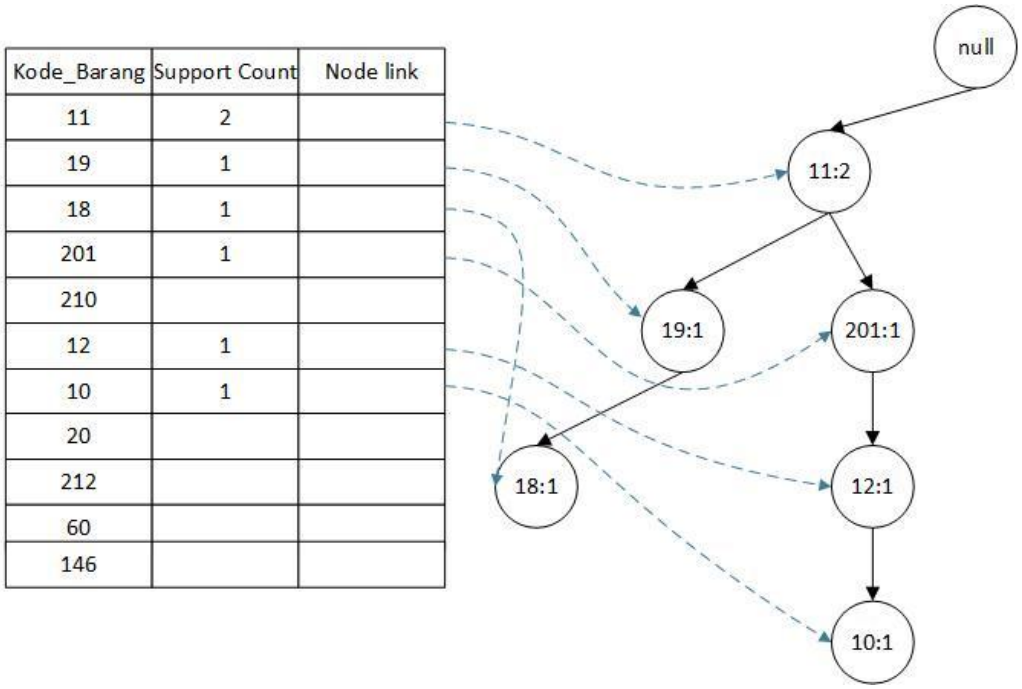
Kode_Barang	Support Count	Node link
11	1	
19	1	
18	1	
201		
210		
12		
10		
20		
212		
60		
146		



Gambar 3. 1 Hasil pembentukan *FP-Tree* faktur 1101

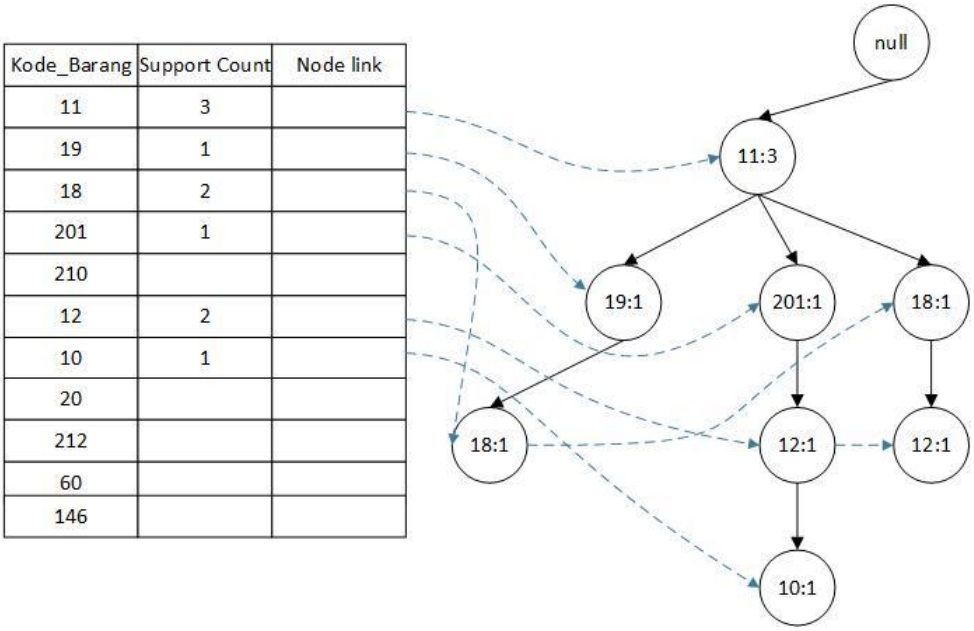
Berikut ini merupakan pembentukan *FP-Tree* untuk faktur 1101 dengan produk yang dipesan adalah 11,201,12,10:

Kode_Barang	Support Count	Node link
11	2	
19	1	
18	1	
201	1	
210		
12	1	
10	1	
20		
212		
60		
146		



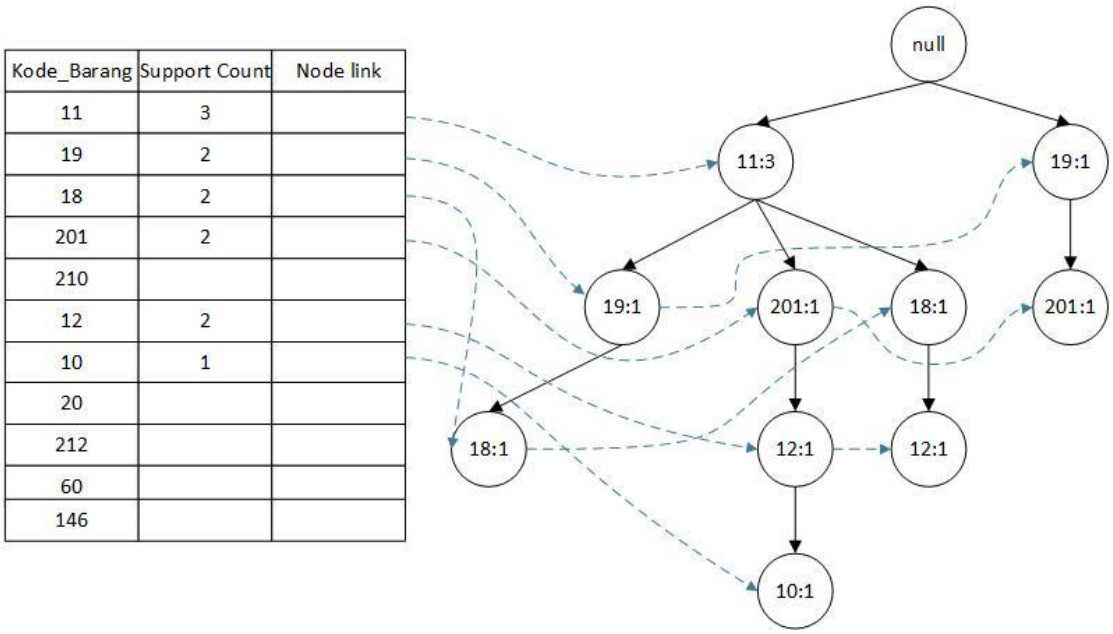
Gambar 3. 2 Hasil pembentukan *FP-Tree* faktur 1102

Berikut ini merupakan hasil pembentukan *FP-Tree* faktur 1103 dengan barang yang dipesan adalah 11,18,12 :



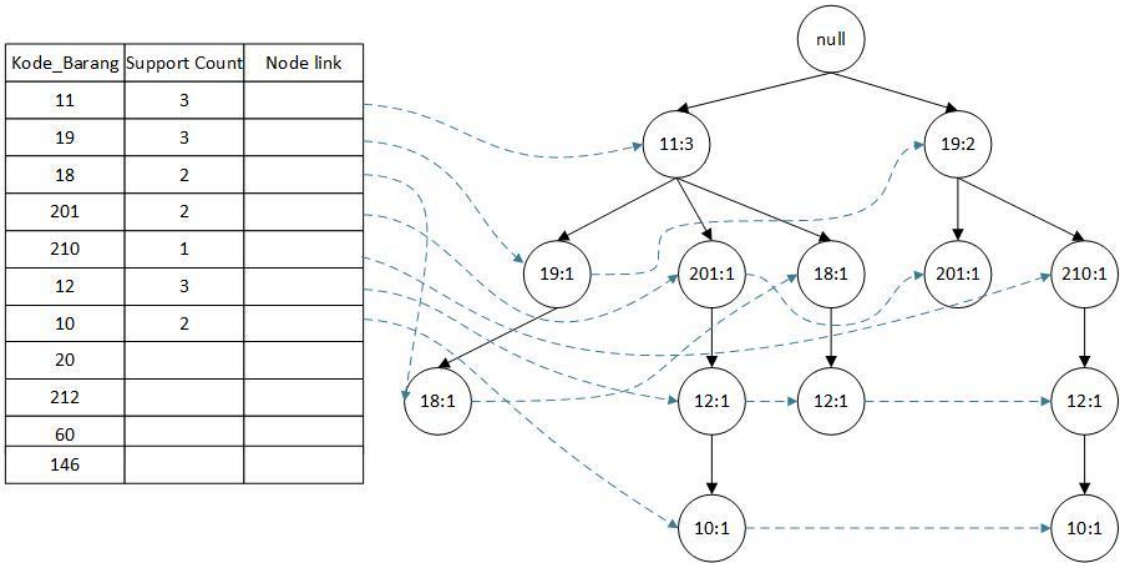
Gambar 3. 3 Hasil pembentukan *FP-Tree* faktur 1103

Berikut ini merupakan hasil pembentukan *FP-Tree* faktur 1104 dengan barang yang dipesan adalah 19, 201:



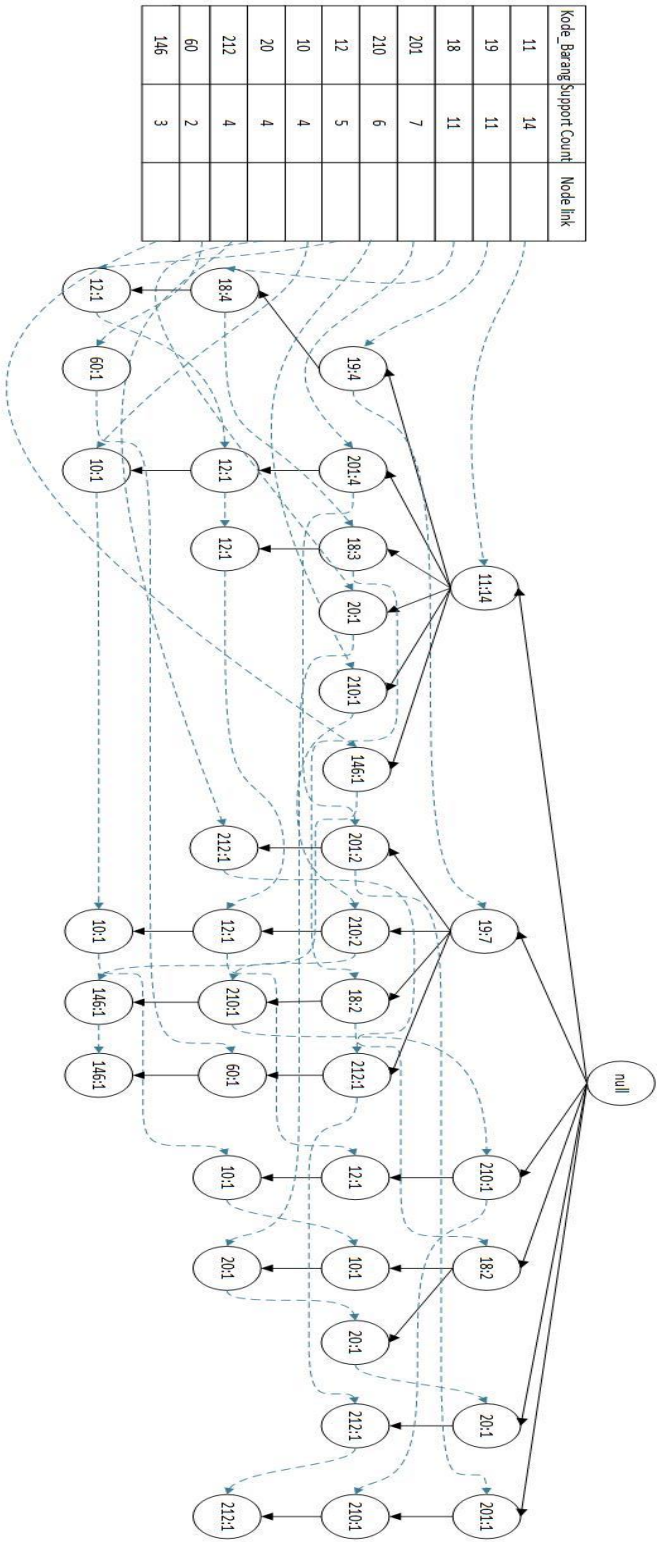
Gambar 3. 4 Hasil pembentukan *FP-Tree* faktur 1104

Berikut ini merupakan hasil pembentukan *FP-Tree* faktur 1105 dengan barang yang dipesan adalah 19,210,12,10 :



Gambar 3. 5 Hasil pembentukan *FP-Tree* faktur 1105

Setelah proses pembacaan faktur 1129 yaitu 19,18,210,146 maka terbentuklah *Tree* dari semua transaksi yang akan digunakan seperti pada Gambar 3. 6 Hasil pembentukan *FP-Tree* faktur 1129:



Gambar 3. 6 Hasil pembentukan *FP-Tree* faktur 1129

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 : 171, 146, 60 40, 212, 10, 20, 12, 210, 18, 19, 201, 11. Berikut adalah hasil tabel dari *conditional pattern base :*

Tabel 3. 6 *Conditional pattern base*

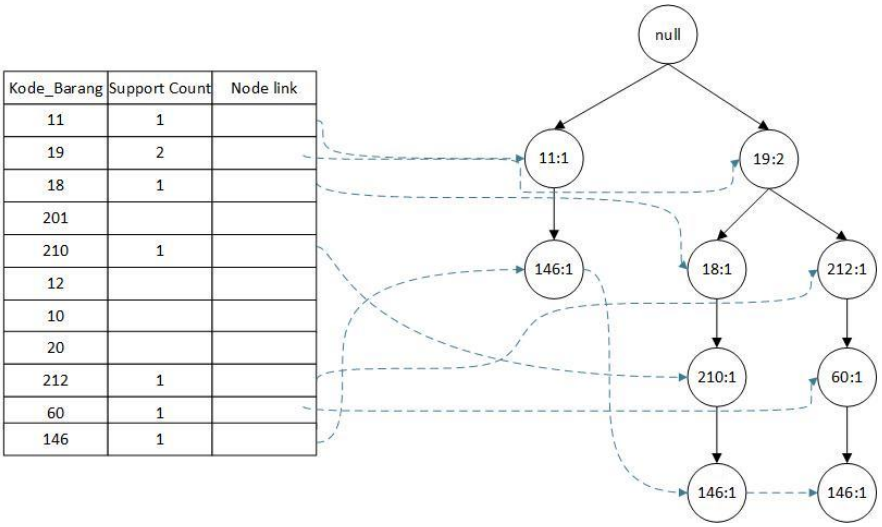
Item	***Conditonal Pattern Base***
146	{11 : 1} {19,18,210:1} {19,212,60:1}
60	{11,19,18,60:1} {19,212:1}
212	{19,201:1} {19:1} {20:1} {201.210:1}
20	{11:1} {18,10:1} {18:1}
10	{11,201,12:1} {19,210,12:1} {210,12:1} {18:1}
12	{11,19,18:1} {11,201:1}

Item	*Conditonal Pattern Base*
	{11,18:1} {19,210:1} {210:1}
210	{11:1} {19:2} {19,18:1} {201:1}
201	{11:4} {19:2}
18	{11,19:4} {11:3} {19:2}
19	{11:4}

b. Tahap pembangkitan *Conditional FP-Tree*

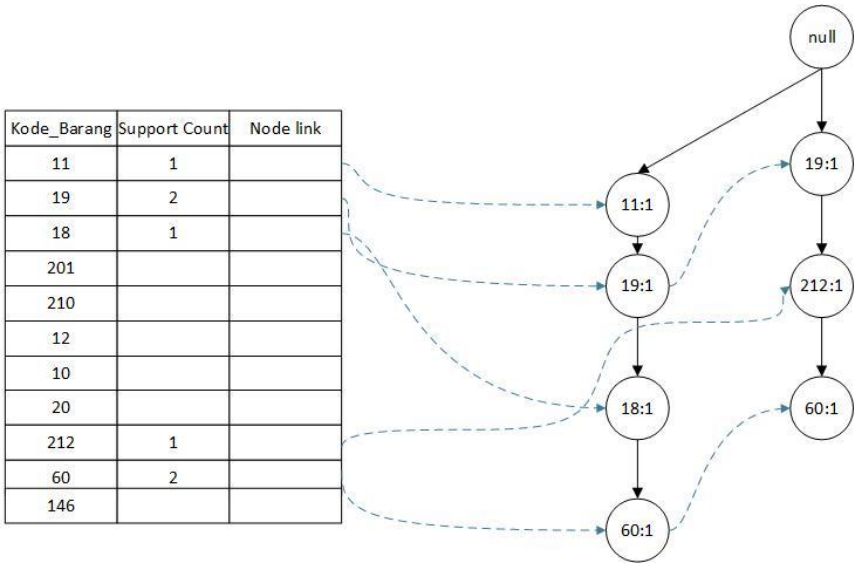
Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support* akan dibangkitkan dengan *Conditional FP-Tree.*

1. *Conditional FP-Tree* untuk kode produk 146 yaitu *null* karena walaupun *support*nya memenuhi nilai *support*, tetapi pada proses pembangkitan tidak didapatkan support count yang memenuhi nilai *minimum support*



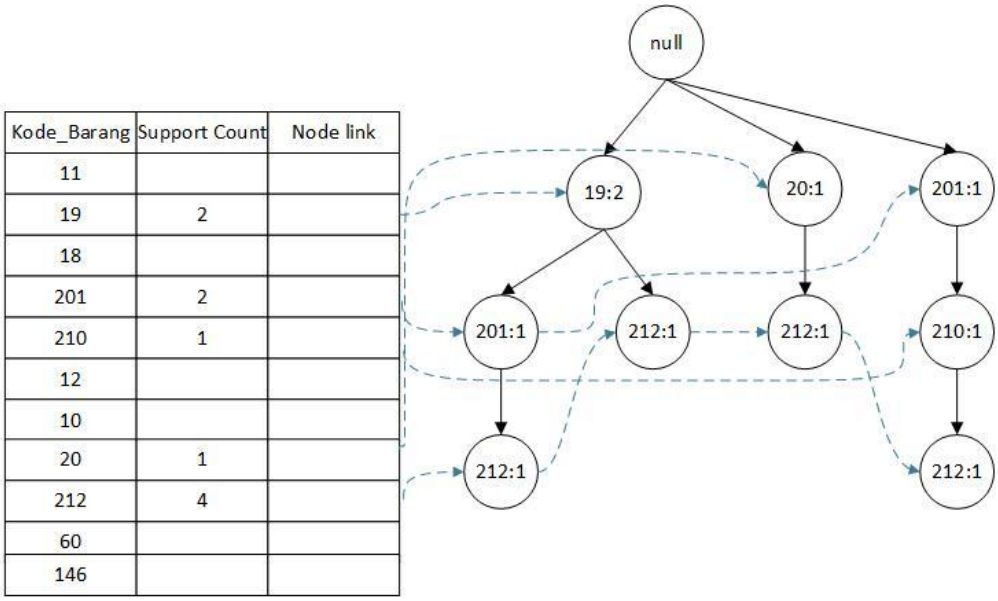
Gambar 3. 7 Lintasan yang mengandung kode barang 146

2. *Conditional FP-Tree* untuk kode produk 60 yaitu *null* karena walaupun *support*nya memenuhi nilai *support*, tetapi pada proses pembangkitan tidak didapatkan support count yang memenuhi nilai *minimum support*



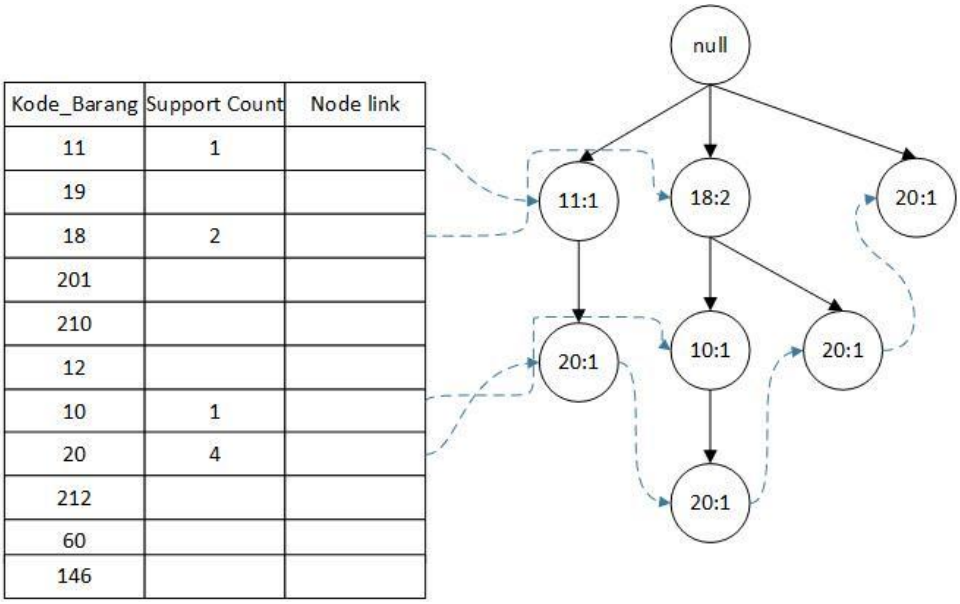
Gambar 3. 8 Lintasan yang mengandung kode barang 60

3. Conditional FP-Tree untuk kode produk 212 yaitu *null* karena walaupun *support*nya memenuhi nilai *support*, tetapi pada proses pembangkitan tidak didapatkan support count yang memenuhi nilai *minimum support*



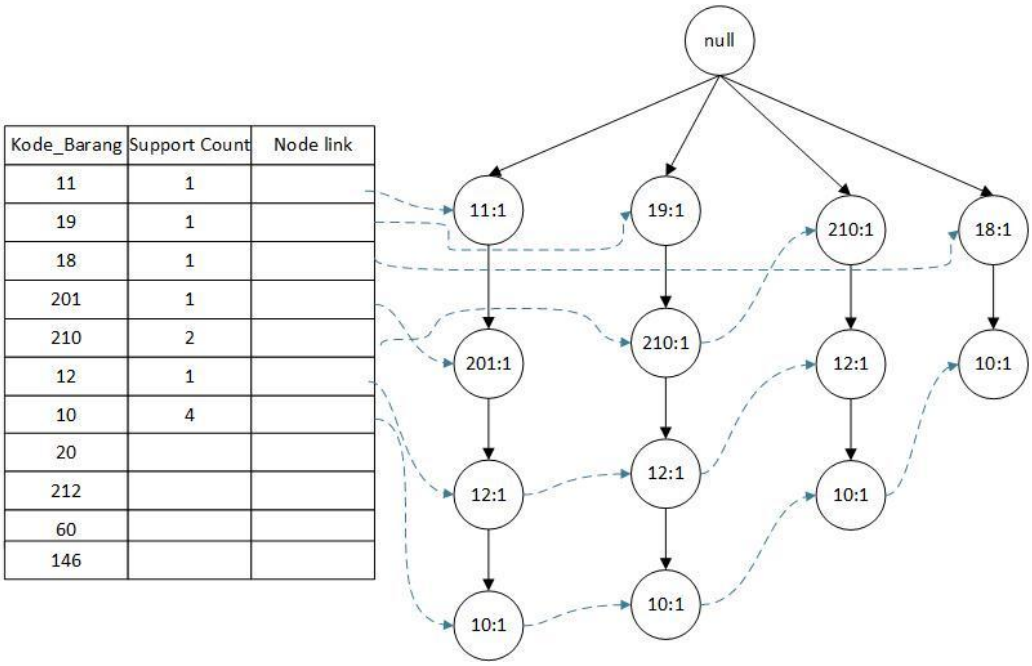
Gambar 3. 9 Lintasan yang mengandung kode barang 212

4. *Conditional FP-Tree* untuk kode produk 20 yaitu *null* karena walaupun *support*nya memenuhi nilai *support*, tetapi pada proses pembangkitan tidak didapatkan support count yang memenuhi nilai *minimum support*



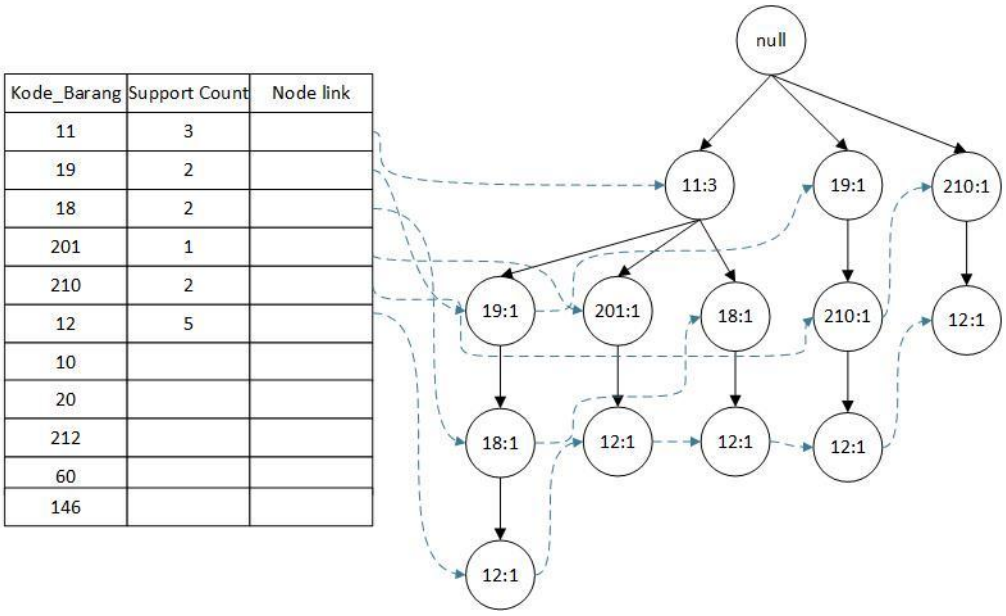
Gambar 3. 10 Lintasan yang mengandung kode barang 20

5. *Conditional FP-Tree* untuk kode produk 10 yaitu *null* karena walaupun *support*nya memenuhi nilai *support*, tetapi pada proses pembangkitan tidak didapatkan support count yang memenuhi nilai *minimum support*



Gambar 3. 11 Lintasan yang mengandung kode barang 10

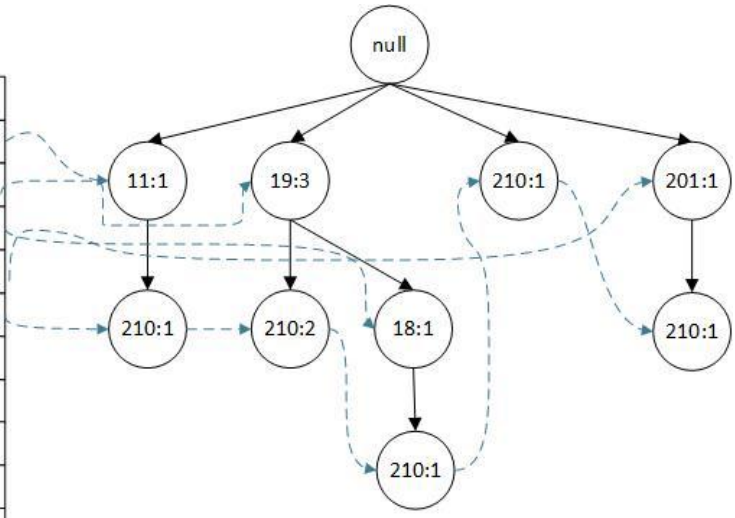
6. *Conditional FP-Tree* untuk kode produk 12 {11:3}



Gambar 3. 12 Lintasan yang mengandung kode barang 12

7. *Conditional FP-Tree* untuk kode produk 210 {19:3}

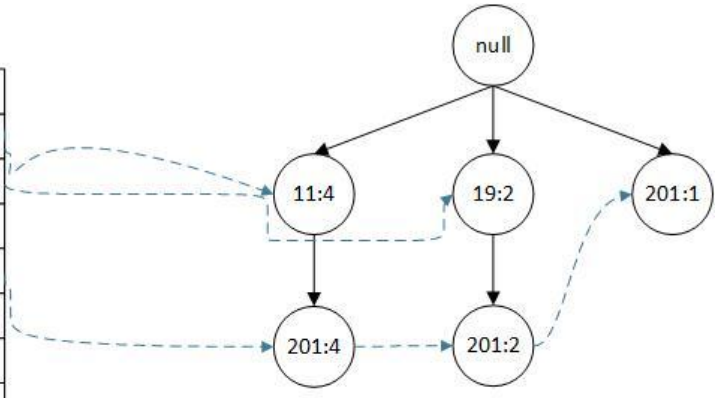
Kode_Barang	Support Count	Node link
11	1	
19	3	
18	1	
201	1	
210	6	
12		
10		
20		
212		
60		
146		



Gambar 3. 13 Lintasan yang mengandung kode barang 210

8. *Conditional FP-Tree* untuk kode produk 201 {11:4}

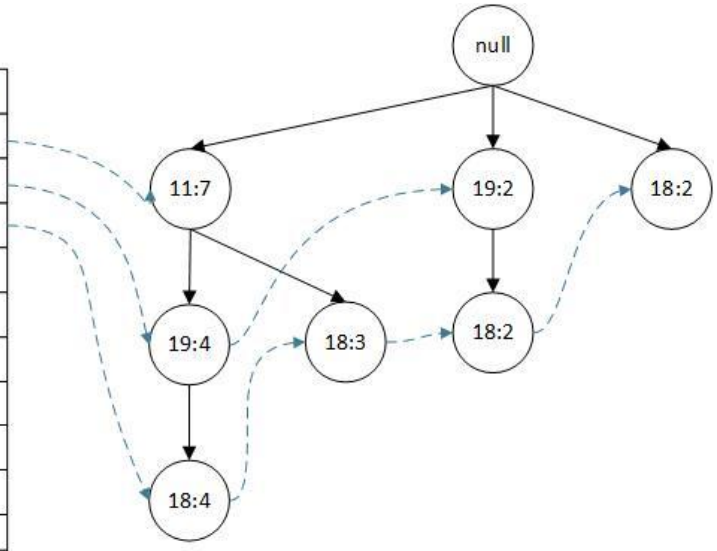
Kode_Barang	Support Count	Node link
11	4	
19	2	
18		
201	7	
210		
12		
10		
20		
212		
60		
146		



Gambar 3. 14 Lintasan yang mengandung kode barang 201

9. *Conditional FP-Tree* untuk kode produk 18 {11:7} {19:6}

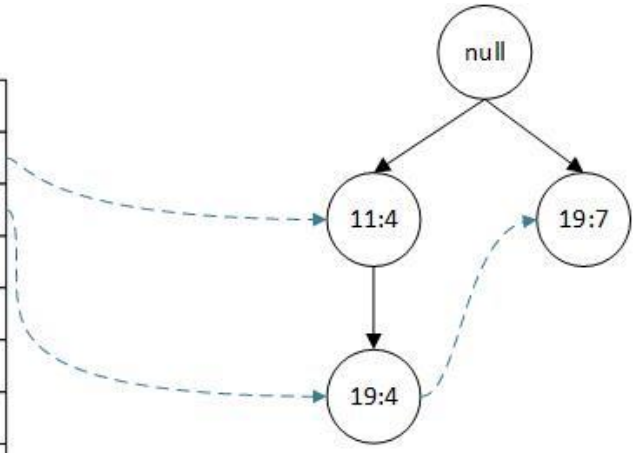
Kode_Barang	Support Count	Node link
11	7	
19	6	
18	11	
201		
210		
12		
10		
20		
212		
60		
146		



Gambar 3. 15 Lintasan yang mengandung kode barang 18

10. *Conditional FP-Tree* untuk kode produk 19 {11:4}

Kode_Barang	Support Count	Node link
11	4	
19	11	
18		
201		
210		
12		
10		
20		
212		
60		
146		



Gambar 3. 16 Lintasan yang mengandung kode barang 19

Setelah mendapatkan hasil conditional *fp-tree*, ditemukan beberapa *frequent itemset* yang keluar seperti pada Tabel 3. 7 *Conditional FP-Tree*:

Tabel 3. 7 *Conditional FP-Tree*

Item	***Conditional FP-tree***
12	{11:3}
210	{19:3}
201	{11:4}
18	{11:7} {19:6} {11,19:4}
19	{11:4}

Setelah mendapatkan hasil *conditional fp-tree* dari tiap *frequent itemset*, selanjutnya adalah menentukan *suffix* dari tiap *frequent itemset*, seperti pada Tabel 3. 8 Hasil *frequent* :

Tabel 3. 8 Hasil *frequent pattern*

Suffix	***Frequent Pattern***
12	{12,11}
210	{210,19}
201	{201,11}
18	{18,11} {18,19} {18,11,19}
19	{19,11}

Dari hasil *frequent pattern* yang didapatkan pada Tabel 3. 8 langkah selanjutnya yaitu mencari *subset* dari masing-masing kode_barang, seperti pada Tabel 3. 9:

Tabel 3. 9 Pembentukkan *subset*

kode_barang	***Subset***
12	{12,11} {11,12}

kode_barang	***Subset***
210	{210,19} {19,210}
201	{201,11} {11.201}
18	{18,11} {11,18} {18,19} {19,18} {18,11,19} {11,18,19} {19,18,11} {19,11,18} {18,19,11} {11,19,18}
19	{19,11} {11,19}

Setelah *frequent pattern* dan *subset* diketahui maka langkah selanjutnya adalah penentuan *anticendent* dan *consequent.* Misalkan:

$$A \rightarrow B$$

maka A akan menjadi *anticendent* dan B akan menjadi *consequent.*

Setelah penentuan *anticendent* dan *consequent* dilakukan makan langkah selanjutnya akan dilakukan proses *generate* untuk mendapatkan *rule* dengan *minimum support* 3 dan *minimum confidence* 60% dengan masing-masing *itemset* dikombinasikan dengan *itemset* lain. Dengan menggunakan rumus yang ada pada bagian bab 2 untuk memperoleh nilai *supporrt* dan nilai *confidence*. Berikut hasil perhitungan *minimum support* dan *minimum confidence*.

Tabel 3. 10 Hasil perhitungan *confidence*

Rule	***Minimum Confidence***
12 → 11	(3/5)*100%=60%
11 → 12	(3/15)*100%=20%
210 → 19	(3/7)*100%=43%
19 → 210	(3/9)*100%=33%
18 → 11	(7/11)*100%=64%
18 → 19	(6/11)*100%=55%
18 → 11,19	(4/11)*100%=36%
11→ 18,19	(4/15)*100%=27%
19 → 18,11	(4/11)*100%=37%
18,11 → 19	(4/7)*100%=57%
18,19 → 11	(4/6)*100%=67%
19,11 → 18	(4/4)*100%=100%
11 → 18	(7/15)*100%=47%
19 → 18	(6/11)*100%=55%
19 → 11	(4/11)*100%=37%
11 → 19	(4/15)*100%=27%

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

Tabel 3. 11 Paket barang dalam bentuk awal

Rule	***Minimum Confidence***
12 → 11	60%
18 → 11	64%
18,19 → 11	67%
19,11 → 18	100%

Setelah didapatkan bentuk awal paket barang, maka tahap selanjutnya adalah dilakukan proses *lookup* ke tabel barang yang terdapat pada lampiran D tabel D-5. Hasil *lookup* ke tabel barang sebagai berikut :

Tabel 3. 12 Hasil pemakettan *rules*

Rule	**Paket produk**
12 → 11	Jika Indomie goreng spesial dibeli, maka Indomie ayam bawang dibeli dengan confidence 60%
18 → 11	Jika ABC kopi Sachet dibeli, maka Indomie ayam bawang dibeli dengan confidence 64%
18,19 → 11	Jika ABC kopi Sachet dan Kopi luwak Sachet dibeli, maka Indomie ayam bawang dibeli dengan confidence 67%
19,11 → 18	Jika Kopi luwak Sachet dan Indomie aya bawang dibeli, maka ABC kopi Sachet dibeli dengan confidence 100%

Dalam penentuan kombinasi barang yang akan dijual dalam bentuk paket pihak Primer Koperasi Kartika memiliki batasan khusus, yaitu :

1. Harga perpaket didapatkan dari pengurangan ke ribuan atau puluhribuan terdekat. Misalkan jumlah harga dari paket 1 sebesar Rp.33.200 maka akan dikurangi menjadi Rp. 30.000.
2. Harga perpaket minimal sebesar Rp.15.000 dan maksimal Rp. 60.000.
3. Jumlah barang yang dapat dipaketkan yaitu sebanyak 3 sampai 5 barang.
4. Jumlah paket yang tersedia dalam 1 periode yaitu sebanyak 4 sampai 5 paket.

Dari batasan yang dimiliki oleh Primer Koperasi Kartika maka didapatkan rekomendasi paket sebagai berikut :

Tabel 3. 13 Informasi paket

Paket	Paket produk
Paket 1	1. ABC Kopi Sachet 2. Kopi Luwak Sachet 3. Indomie Ayam bawang
Paket 2	1. Kopi Luwak Sachet 2. Indomie ayam bawang 3. ABC Kopi Sachet

3.1.3 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*.

3.1.3.1 Analisis Kebutuhan Perangkat Keras

Analisis spesifikasi kebutuhan perangkat keras bertujuan untuk mengetahui spesifikasi perangkat keras yang digunakan. Berdasarkan pengamatan yang dilakukan di Primer Koperasi Kartika, spesifikasi perangkat keras yang digunakan oleh Primer Koperasi Kartika adalah sebagai berikut:

Tabel 3. 14 Spesifikasi 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. 15 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 Primer Koperasi Kartika seperti pada tabel 3.25 dan tabel spesifikasi perangkat keras yang dibutuhkan seperti pada tabel 3.26 maka dapat disimpulkan bahwa semua perangkat keras yang dibutuhkan telah memadai untuk menjalankan sistem.

3.1.3.2 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 Primer Koperasi Kartika 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 Primer Koperasi Kartika, spesifikasi perangkat lunak yang digunakan masih kurang. Untuk mengatasi hal tersebut akan di install MySQL dan juga Visual Studio pada komputer yang digunakan.

3.1.3.3 Analisis Kebutuhan Pengguna

User atau pengguna yang akan mengoperasikan aplikasi ini merupakan manager yang bertindak sebagai pengolah bisnis. Analisis Pengguna dapat dilihat pada tabel 3.27

Tabel 3. 16 Analisis pengguna

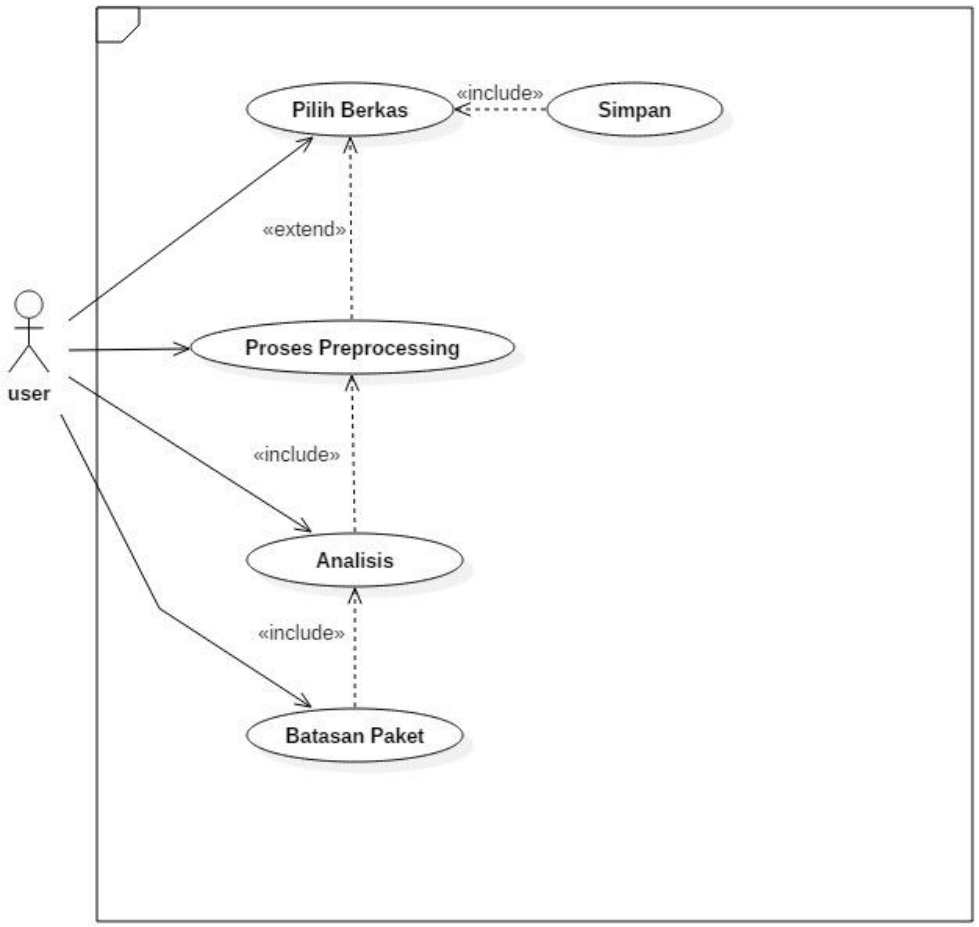
User	**Akses**	**Keahlian**
Manager	Manager dapat mengolah data transaksi penjualan dan memperoleh informasi dari aplikasi *mining*.	Dapat mengoperasikan aplikasi dengan baik seperti penginputan data transaksi, penghapusan data transaksi, dan juga dapat mengolah informasi yang didapat.

3.1.4 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 *usecase*, *activity diagram*, *sequence diagram* dan c*lass diagram.*

3.1.4.1 Diagram Usecase

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 5 *use case*. Adapun diagram *use case* dari sistem yang akan dibuat dapat dilihat di Gambar 3. 17 Diagram Use Case Pada Sistem Data Mining:



Gambar 3. 17 Diagram Use Case Pada Sistem Data Mining

Tabel 3. 17 Tabel Definisi User

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

Tabel 3. 18 Deskripsi Use Case

No	Use Case	Deskripsi
1	Pilih berkas	Proses pemilihan berkas adalah proses untuk memilih berkas yang akan dimasukkan ke dalam program.
2	Simpan	Sistem melakukan proses penyimpanan data transaksi yang sudah dipilih ke dalam *data base*.
3	Proses preprocessing	Sistem melakukan proses pemilihan, pembersihan, perhitungan frekuensi dan pengurutan data
4	Analisis	Sistem melakukan proses analisis data yang sudah dipilih, sedangkan *user* memasukkan nilai *minimum support* dan *minimum confidence*
5	Batasan paket	*User* memasukkan batasan paket sesuai dengan kebutuhan Primer Koperasi Kartika. Sistem melakukan proses batasan paket.

3.1.4.2 Skenario *Use Case*

Skenario *use case* menjelaskan skenario dari setiap proses yang digambarkan pada diagram *use case*. Berikut ini skenario use case dari Gambar 3. 17 Diagram Use Case Pada Sistem Data Mining:

1. Skenario *Use Case* Pilih Berkas

Skenario *use case* pilih berkas menggambarkan langkah – langkah aksi *user* dan sistem melakukan proses pemilihan data yang akan digunakan.

Tabel 3. 19 Requirement A.1

Requirement A.1
Sistem menyediakan menu pilih berkas untuk memilih file mana yang akan dipilih untuk dilakukan proses selanjutnya.

Tabel 3. 20 Skenario Use Case Pilih Berkas

Use case Name	**Pilih Berkas**	
Related Requirements	Requirement *A.1*	
Goal In Context	Memilih *file* yang akan di simpan dalam *data base* berhasil	
Precondition	*User* memilih data	
Successful End Condition	Data transaksi yang dipilih berhasil ditampilkan ke dalam *data grid*	
Failed End Condition	Data gagal ditampilkan dalam *grid.*	
Actors	*User*	
Triger	*User* memilih data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*User* memilih menu pilih berkas
	2	Sistem menampilkan dialog box
	3	*User* memilih data yang akan di masukkan
	4	Sistem melakukan proses validasi format *file*
	5	Sistem menampilkan data di *gridview*
Extension	Step	Branching Action
	4.1	Menampilkan pesan kesalahan format *file* dalam proses pemilihan data

2. Skenario *Use Case* Simpan

Skenario *use case* simpan menjelaskan langkah-langkah sistem untuk melakukan proses penyimpanan data ke *data base*

Tabel 3. 21 Requirement A.2

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

Tabel 3. 22 Skenario Use Case Simpan

Use case Name	**Simpan**
Related Requirements	Requirement *A.1*, Requirement *A.2*
Goal In Context	Sistem melakukan proses penyimpanan data yang sudah terpilih ke *data base*
Precondition	Data transaksi sudah ditampilkan dalam grid
Successful End Condition	Sistem berhasil melakukan proses penyimpanan data ke *data base*

Failed End Condition	Sistem gagal untuk melakukan proses penyimpanan data ke *data base*	
Actors	*User*	
Triger	*User* melakukan *request* proses simpan data kepada sistem	
Included Cases	-	
Main Flow	Step	Action
	1	*User* memilih menu simpan
	2	Sistem melakukan validasi atribut tabel
	3	Sistem melakukan proses simpan
	4	Sistem menampilkan pesan data berhasil tersimpan
Extension	Step	Branching Action
	2.1	Sistem menampilkan pesan bahwa data gagal disimpan karena atribut tabel berbeda

3. Skenario *Use Case* Proses Preprocessing

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

Tabel 3. 23 Requirement A.3

Reqirement A.3
Sistem menyediakan menu *preprocessing* untuk melakukan proses preprocessing, seperti pemilihan atribut, penghapusan, perhitungan frekuensi dan pengurutan data sesuai dengan frekuensi.

Tabel 3. 24 Skenario Use Case Proses Preprocessing

Use case Name	**Proses Preprocessing**
Related Requirements	Requirement *A.1*, Requirement *A.2*, Requirement *A.3*
Goal In Context	Sistem berhasil melakukan proses preprocessing
Precondition	Data sudah berhasil tersimpan di *data base*
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 precosessing	
Included Cases	Pemilihan atribut, pembersihan data, hitung frekuensi dan pengurutan data.	
Main Flow	Step	Action
	1	*User* memilih menu proses preprocessing
	2	Sistem melakukan proses pemilihan data
	3	Sistem melakukan proses pembersihan data
	4	Sistem melakukan proses hitung frekuensi
	5	Sistem melakukan proses pengurutan data
	6	Sistem menampilkan semua hasil proses preprocessing dalam grid
Extension	Step	Branching Action
	-	

4. Skenario *Use Case* Asosiasi

Skenario *Use case* asosiasi menggambarkan langkah-langkah aksi *user* dan sistem untuk melakukan proses asosiasi.

Tabel 3. 25 *Requrement* A.4

Reqirement A.4
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. 26 Skenario Use Case Asosiasi

Use case Name	**Analisis**
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 *Minimum Support* dan *Minimum Confidence*	
Main Flow	Step	Action
	1	*User* memasukkan nilai *minimum support* dan *minimum confidence*
	2	*User* memilih menu proses asosiasi
	3	Sistem melakukan validasi terhadap nilai *minimum support* dan *minimum confidence* yang *user* masukkan
	4	Sistem melakukan proses asosiasi
	5	Sistem menampilkan hasil *rules*, nilai support dan confidence tiap item
Extension	Step	Branching Action
	3.1	Sistem menampilkan pesan bahwa nilai *minimum support* dan *minimum confidence* tidak boleh kosong

5. Skenario *Use Case* Batasan Paket

Skenario *Use Case* Batasan Paket menggambarkan langkah-langkah aksi *user* untuk melakukan proses batasan paket.

Tabel 3. 27 *requirement* A.5

Reqirement A.5
Sistem menyediakan menu batasan paket untuk melakukan input parameter berupa nilai maksimum dan minimum dalam satu paket, Eliminasi untuk mengeliminasi item apasaja yang tidak sesuai dengan batasan paket dan rekomendasi paket apa saja yang didapatkan.

Tabel 3. 28 Scenario Use case batasan paket

Use case Name	**Batasan Paket**
Related Requirements	*Requirement A.1, Requirement A.2, Requirement A.3, Requirement* A4

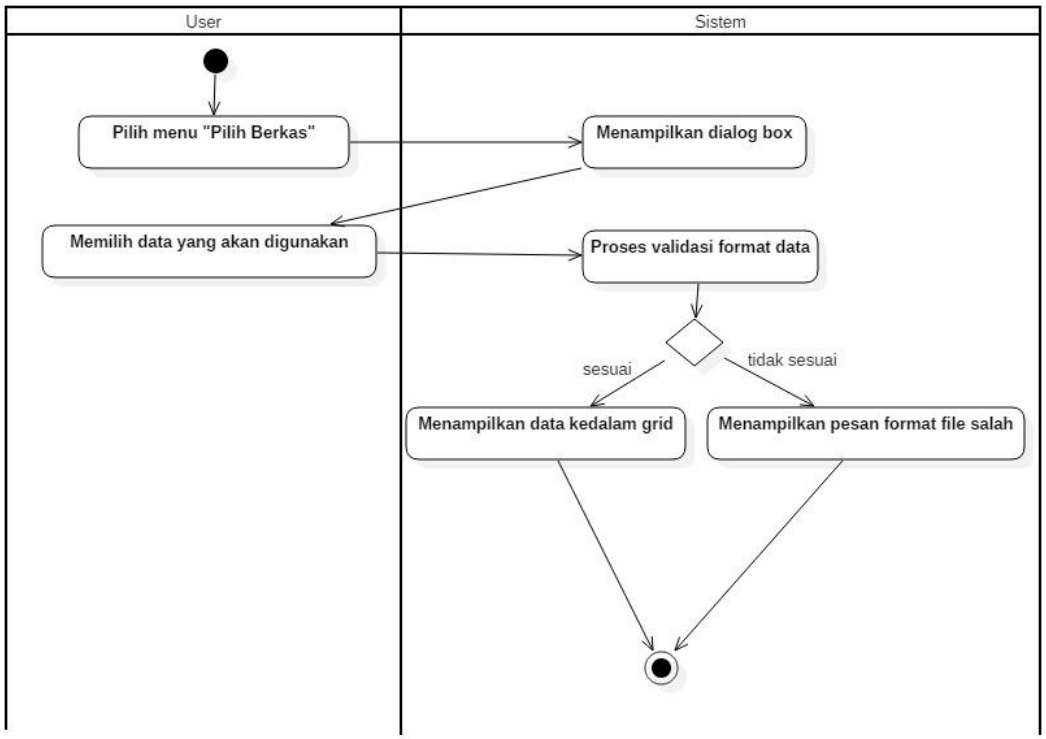
Goal In Context	Sistem menampilkan informasi paket produk yang sudah sesuai dengan batasan	
Precondition	Data sudah diproses pada *use case rules*	
Successful End Condition	Sistem berhasil menampilkan informasi paket yang sesuai dengan batasan paket	
Failed End Condition	Sistem gagal menampilkan informasi paket produk yang sudah seusai	
Actors	*User*	
Triger	*User* melakukan *request* proses batasan paket	
Main Flow	Step	Action
	1	*User* memasukkan nilai harga maksimum dan minimum
	2	*User* memilih menu batasan paket
	3	Sistem melakukan proses batasan paket
	4	Sistem menampilkan hasil proses batasan paket
Extension	Step	Branching Action
	-	

3.1.4.3 *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* Pilih Berkas

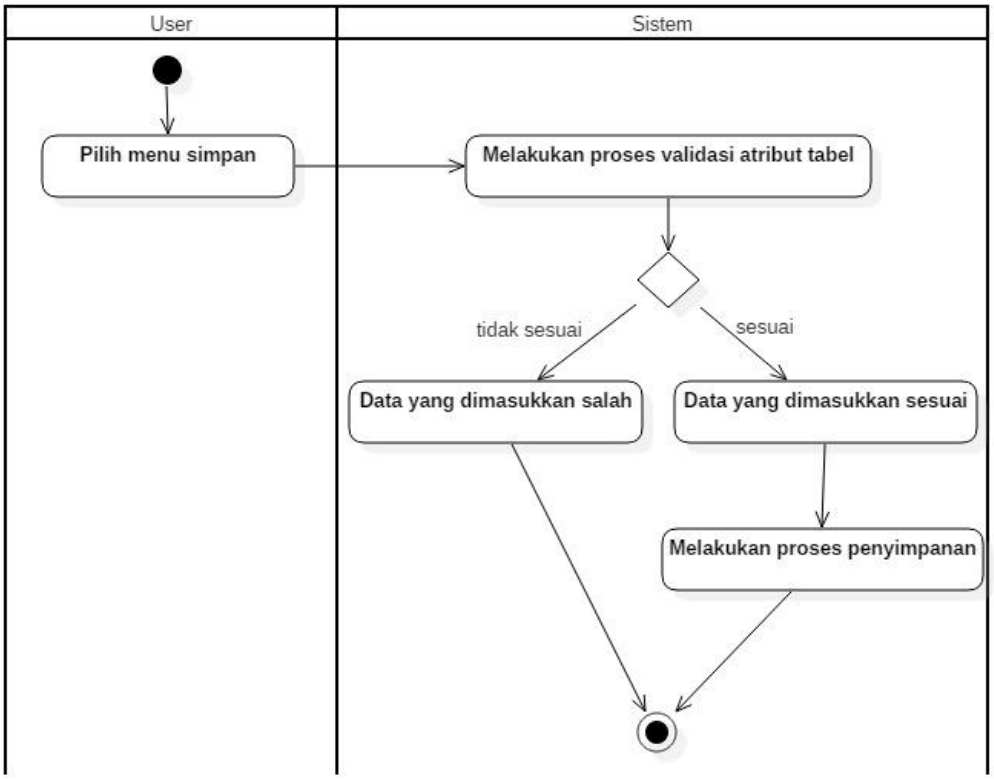
   Activity diagram import data menjelaskan aktivitas *user* dalam melakukan proses pilih berkas. *Activity diagram* pilih berkas dapat dilihat pada Gambar 3. 18 *Activity* diagram



Gambar 3. 18 *Activity* diagram pilih berkas

2. *Activity Diagram* Simpan

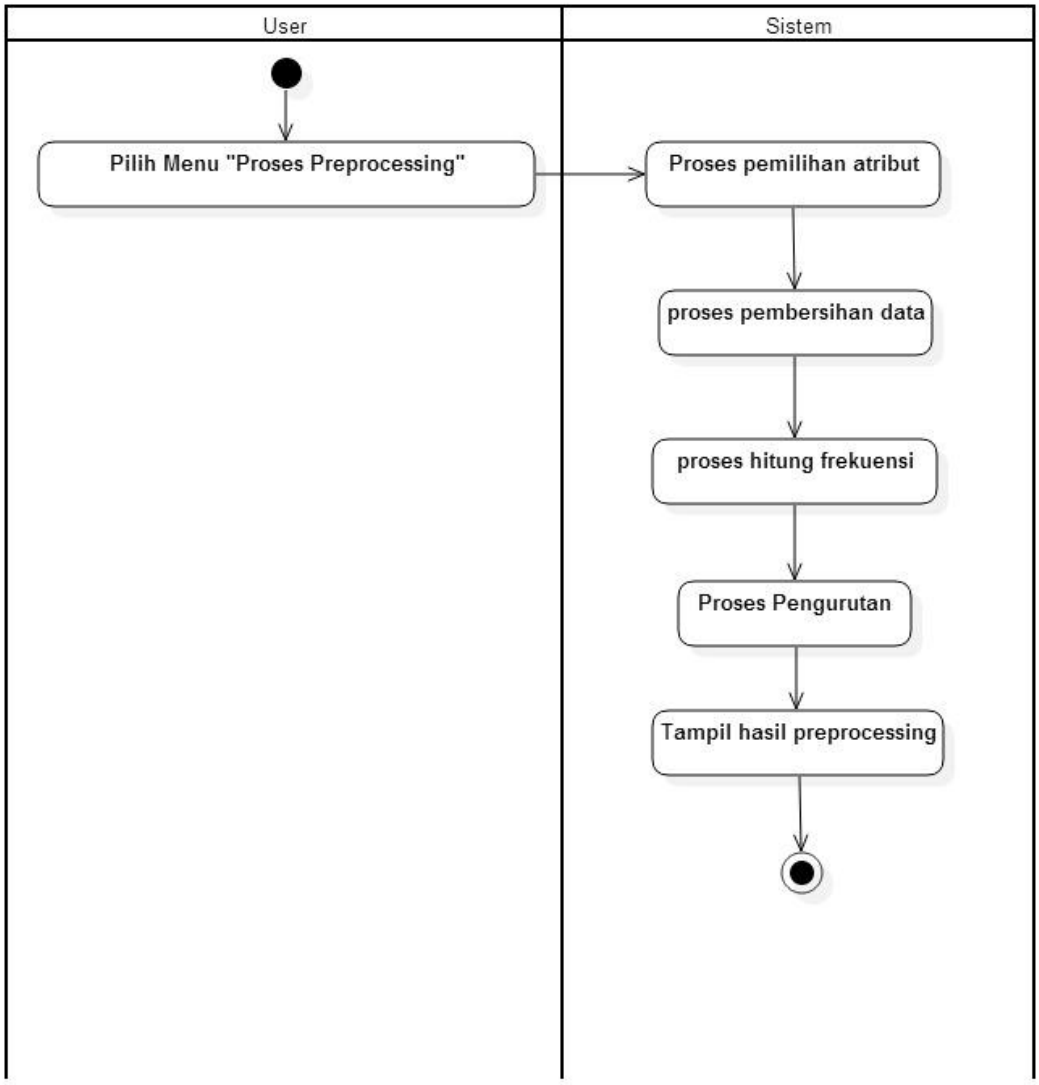
Activity diagram simpan menjelaskan aktivitas sistem dalam melakukan proses penyimpanan data ke *database*. *Activity diagram* simpan dapat dilihat pada Gambar 3. 19



Gambar 3. 19 *Activity diagram* simpan

3. *Activity Diagram* Proses *Preprocessing*

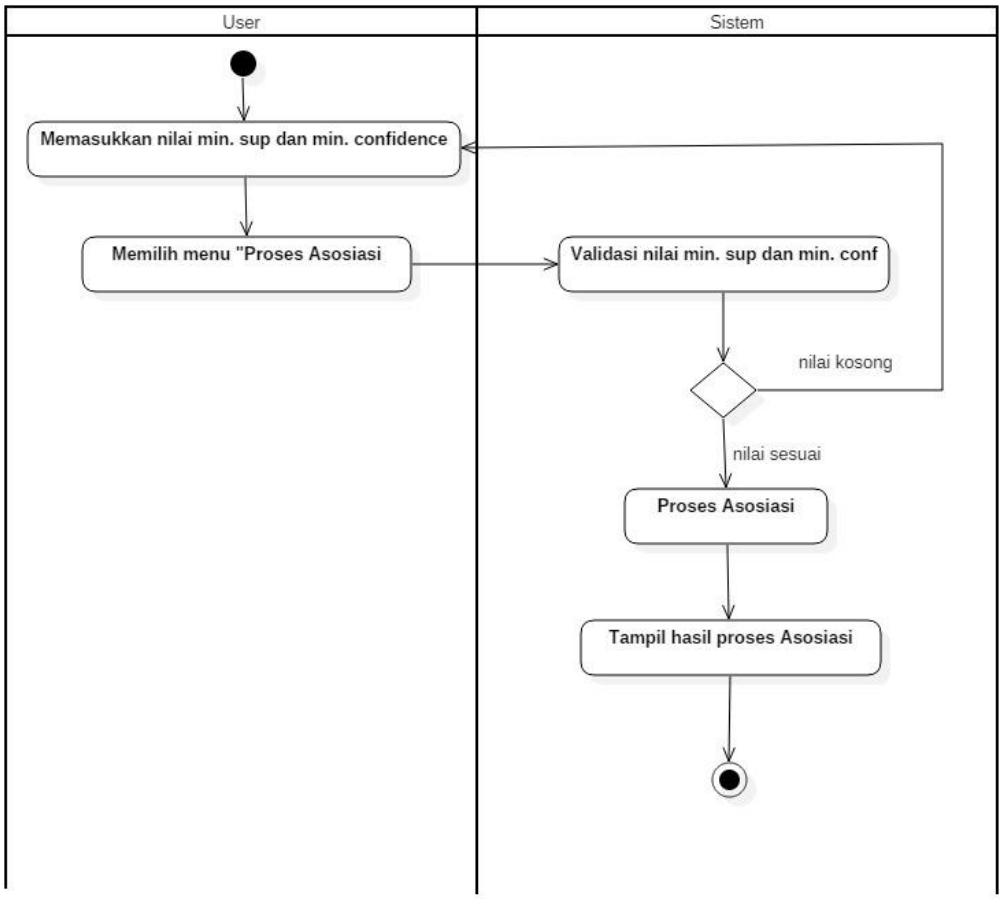
Activity diagram proses *preprocessing* menjelaskan aktivitas *user* dan sistem dalam melakukan proses *preprocessing*. *Activity diagram* proses *preprocessing* dapat dilihat pada Gambar 3. 20



Gambar 3. 20 *Activity diagram* proses *preprocessing*

4. *Activity Diagram* Asosiasi

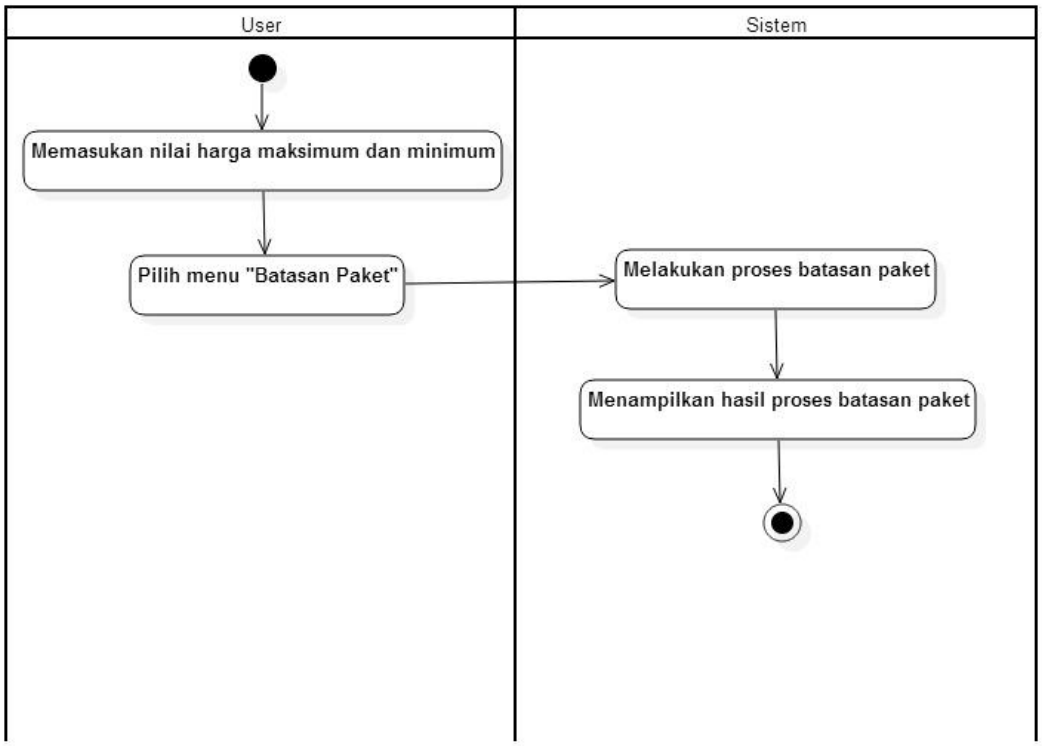
Activity diagram asosiasi menjelaskan aktivitas *user* dan sistem dalam menampilkan hasil seleksi *rule* sesuai nilai *Minimum confidence* yang dimasukkan oleh *user*. *Activity diagram* asosiasi dapat dilihat pada Gambar 3. 21



Gambar 3. 21 *Activity diagram* asosiasi

5. *Activity Diagram* Batasan

Activity diagram batasan menjelaskan aktivitas *user* dan sistem dalam memberikan batasan paket yang sebelumnya dalam bentuk *rule*. *Activity diagram* batasan dapat dilihat pada Gambar 3. 22

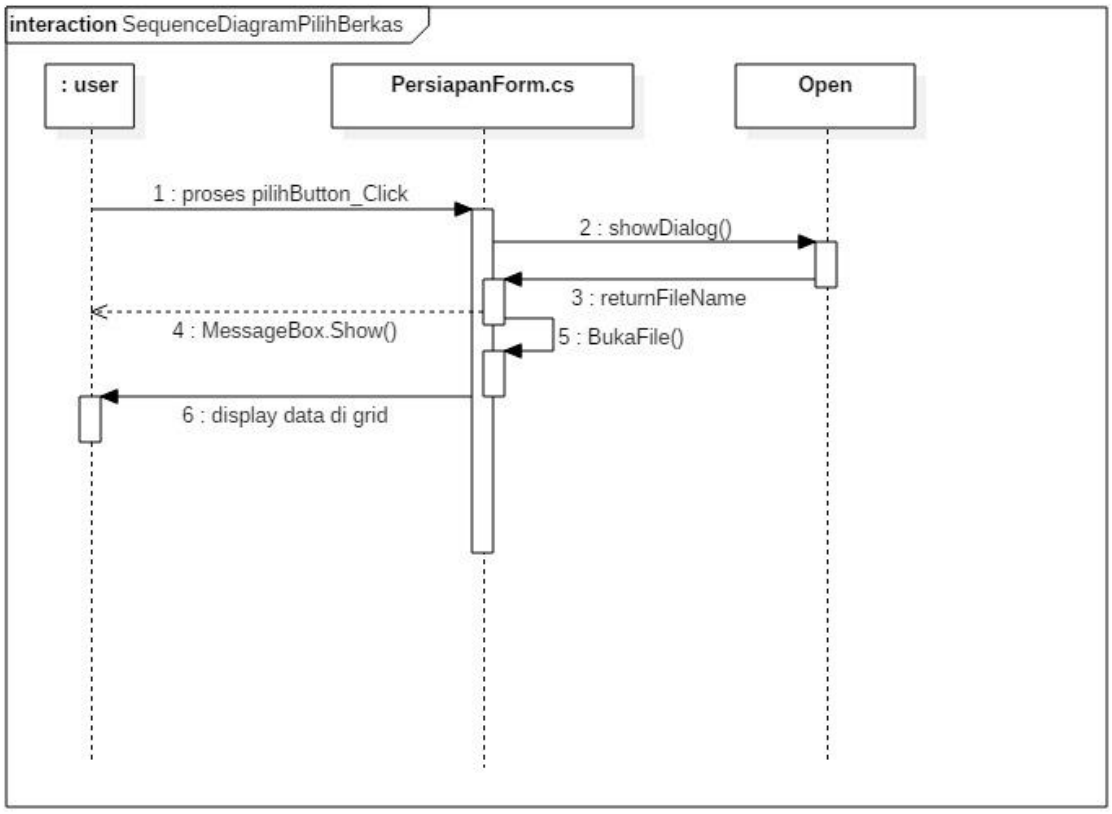


Gambar 3. 22 *Activity Diagram* Batasan

3.1.4.4 *Sequence* Diagram

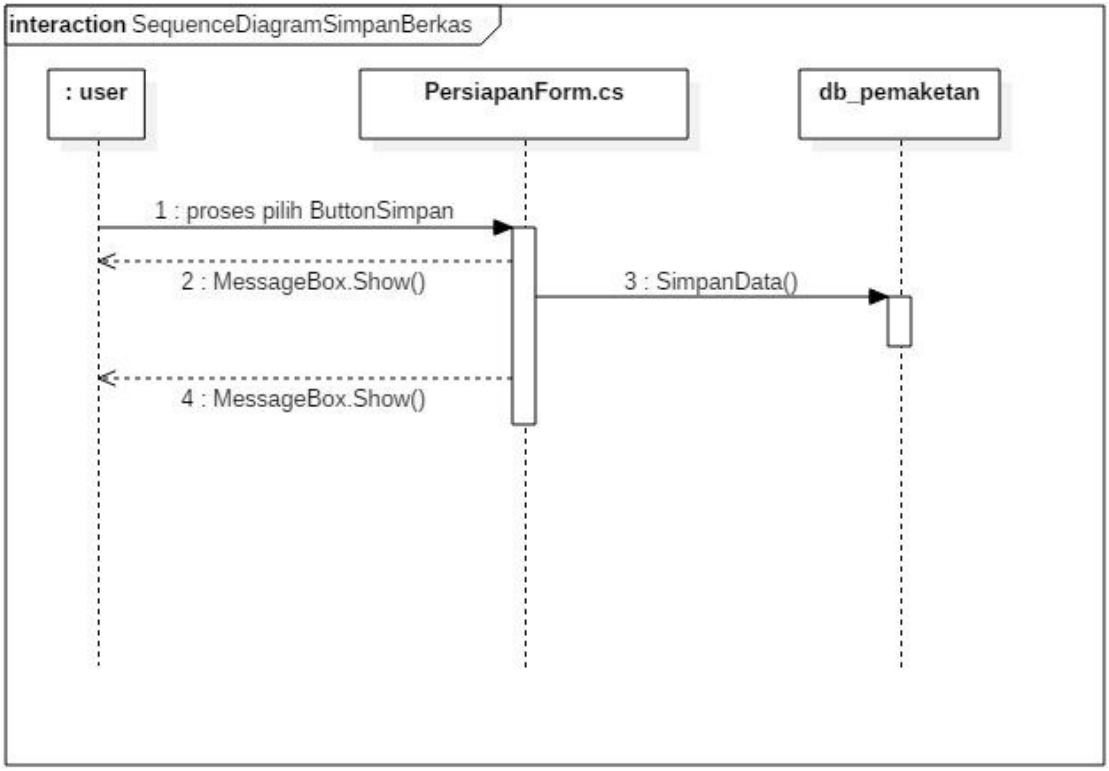
Sequence diagram digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai tujuan hasil. Berikut adalah *sequence diagram* pada aplikasi *data mining* yang sedang dibangun pada Gambar 3. 23 sampai dengan Gambar 3. 27

1. *Sequence Diagram* Pilih Berkas



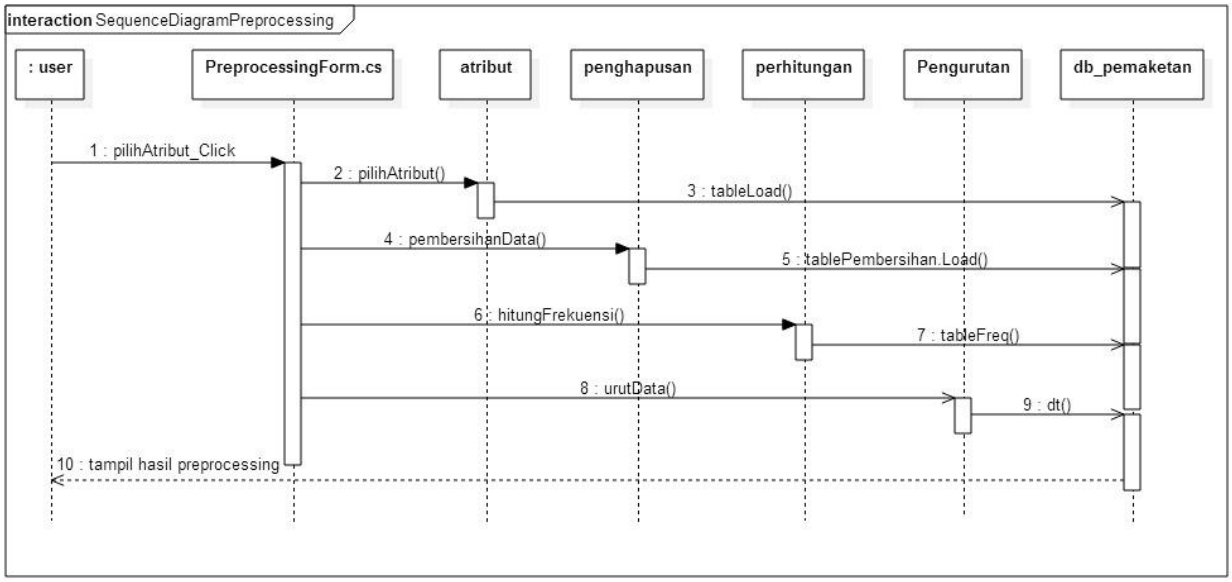
Gambar 3. 23 Sequence diagram pilih berkas

2. *Sequence Diagram* Simpan



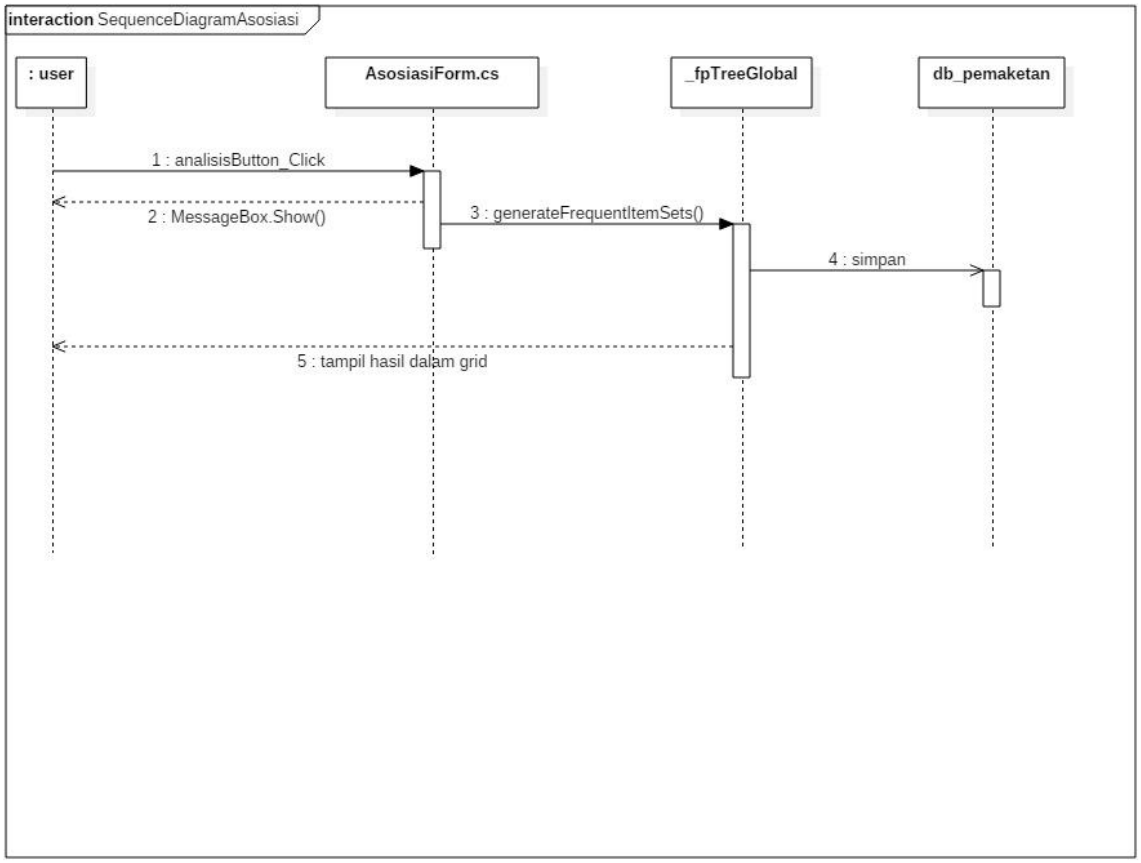
Gambar 3. 24 Sequence diagram simpan

3. *Sequence Diagram* Proses *Preprocessing*



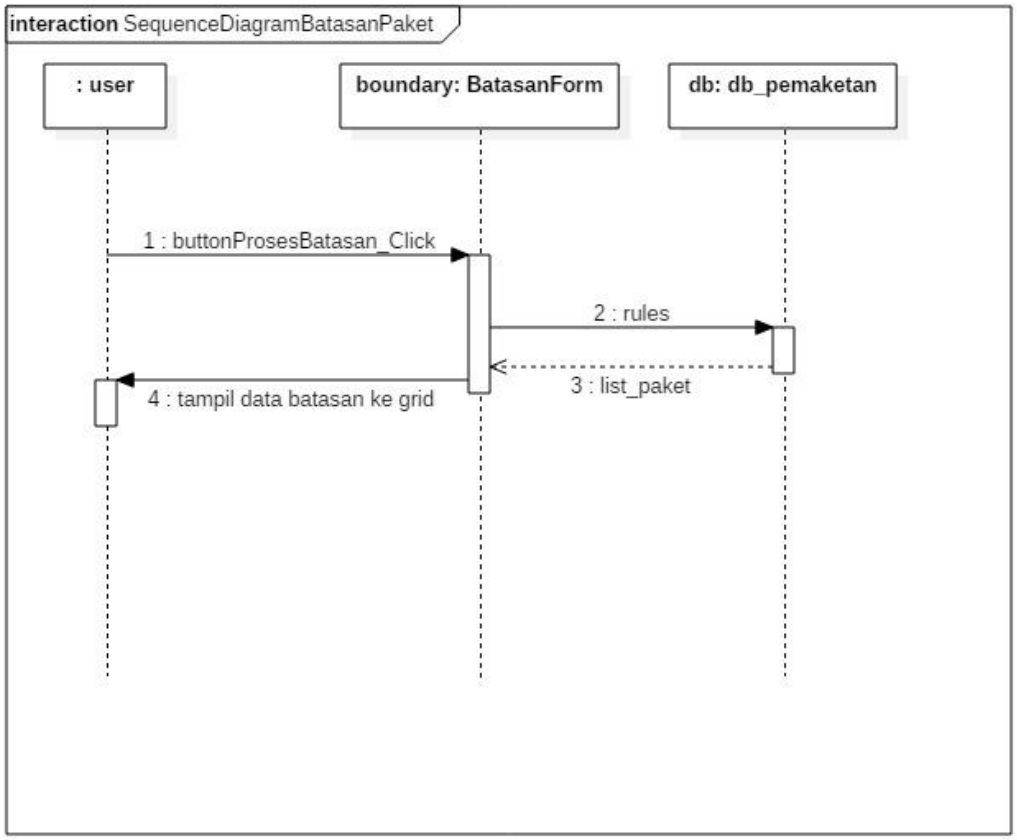
Gambar 3. 25 Sequence diagram proses *preprocessing*

4. *Sequence Diagram* Asosiasi



Gambar 3. 26 Sequence diagram asosiasi

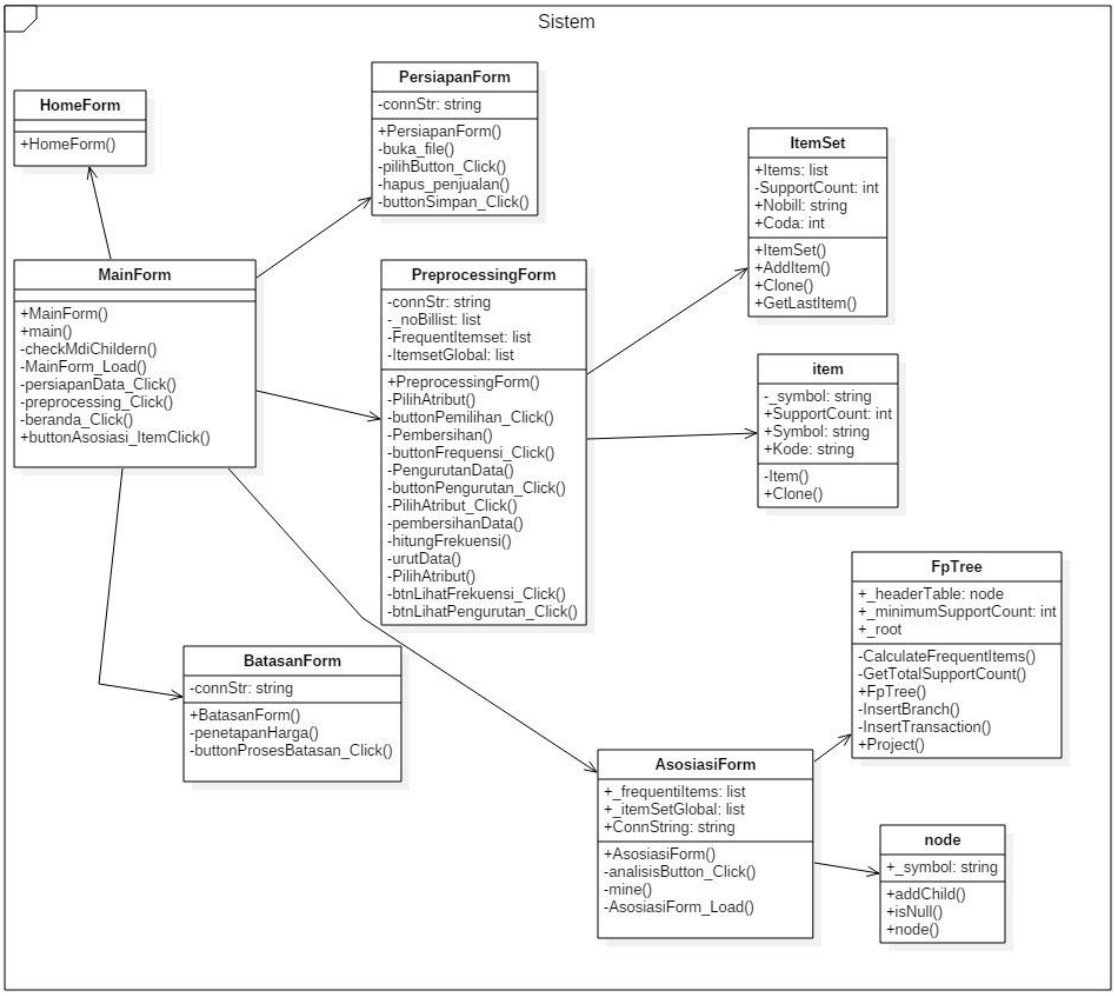
5. *Sequence Diagram* Batasan Paket



Gambar 3. 27 Sequence diagram batasan paket

3.1.4.5 *Class* Diagram

Berikut adalah *class diagram* dari sistem yang akan dibangun. Hasilnya dapat pada Gambar 3. 28



Gambar 3. 28 *Class* Diagram

3.2 Perancangan Sistem

Perancangan sistem berkenaan dengan pembangunan perangkat lunak setelah melaui tahap analisis sistem yang akan dibangun. Beberapa aspek dari perancangan sistem dalam penelitian ini meliputi perancangan *class*, perancangan struktur menu, perancangan antarmuka, perancangan pesan dan jaringan semantik.

3.2.1 Perancangan Class

Berikut ini adalah perancangan class yang digunakan pada sistem yang akan dibangun

Tabel 3. 29 Struktur *class* HomeForm

Nama *class* : HomeForm Jenis *class* : *views*		
Nama	Jenis	*Type*
HomeForm()	Method	Public

Tabel 3. 30 Struktur *class* MainForm

Nama *class* : MainForm Jenis *class* : *views*		
Nama	Jenis	*Type*
MainForm()	Method	Public
Main()	Method	Public
buttonAsosiasi_ItemClick()	Method	Public
checkMdiChildern()	Method	Private
MainForm_Load()	Method	Private
persiapanData_Click()	Method	Private
preprocessing_Click()	Method	Private
beranda_Click()	Method	Private

Tabel 3. 31 Struktur class PersiapanForm

Nama *class* : PersiapanForm Jenis *class* : *views*		
Nama	Jenis	*Type*
connStr: string	Atribut	Private
PersiapanForm()	Method	Public

Nama *class* : PersiapanForm Jenis *class* : *views*		
Nama	Jenis	*Type*
buka_file()	Method	Private
pilihButton_Click()	Method	Private
hapus_penjualan()	Method	Private
buttonSimpan_Click()	Method	Private

Tabel 3. 32 Struktur class PreprocessingForm

Nama *class* : PreprocessingForm Jenis *class* : v*iews*		
Nama	Jenis	*Type*
connStr: string	Atribut	Private
_noBillist: list	Atribut	Private
FrequentItemset: list	Atribut	Private
ItemsetGlobal: list	Atribut	Private
PreprocessingForm()	Method	Public
PilihAtribut_Click()	Method	Private
buttonPemilihan_Click()	Method	Private
Pembersihan()	Method	Private
buttonFrequensi_Click()	Method	Private
PengurutanData()	Mehod	Private
buttonPengurutan_Click()	Method	Private
PilihAtribut()	Method	Private
pembersihanData()	Method	Private
hitungFrekuensi()	Method	Private
urutData()	Method	Private
PilihAtribut()	Mehod	Private
btnLihatFrekuensi_Click()	Method	Private
btnLihatPengurutan_Click()	Method	Private

Tabel 3. 33 Struktur class ItemSet

Nama *class* : ItemSet Jenis *class* : *controller*		
Nama	Jenis	*Type*
Items: list	Atribut	Public
Nobill: string	Atribut	Public
Coda: int	Atribut	Public
SupportCount: int	Atribut	Private
ItemSet()	Method	Public
AddItem()	Method	Public
Clone()	Method	Public
GetLastItem()	Method	Public

Tabel 3. 34 Struktur class item

Nama *class* : item Jenis *class* : *controller*		
Nama	Jenis	*Type*
SupportCount: int	Atribut	Public
Symbol: string	Atribut	Public
Kode: string	Atribut	Public
_symbol: string	Atribut	Private
clone()	Method	Public
item()	Method	Private

Tabel 3. 35 Struktur class AsosiasiForm

Nama *class* : AsosiasiForm Jenis *class* : *views*		
Nama	Jenis	*Type*
_frequentItems: list	Atribut	Public

Nama *class* : AsosiasiForm Jenis *class* : *views*		
Nama	Jenis	*Type*
_itemSetGlobal: list	Atribut	Public
ConnString: string	Atribut	Public
asosiasi()	Method	Public
importData_Load()	Method	Public
initializeComponent()	Method	Public
generateFrequentItemSet()	Method	Public
save_rules()	Method	Public

Tabel 3. 36 Struktur class FpTree

Nama *class* : FpTree Jenis *class* : *controller*		
Nama	Jenis	*Type*
headerTable: node	Atribut	Public
_minimumSupportCount: int	Atribut	Public
_root	Atribut	Public
CalculateFrequentItems()	Method	Public
GetTotalSupportCount()	Method	Public
FpTree()	Method	Public
InsertBranch()	Method	Public
InsertTransaction()	Method	Public
Project()	Method	Public

Tabel 3. 37 Struktur class node

Nama *class* : node Jenis *class* : *controller*		
Nama	Jenis	*Type*
_symbol: string	Atribut	Public
addChild()	Method	Public
isNull()	Method	Public
node()	Method	Public

Tabel 3. 38 Struktur *class* BatasanForm

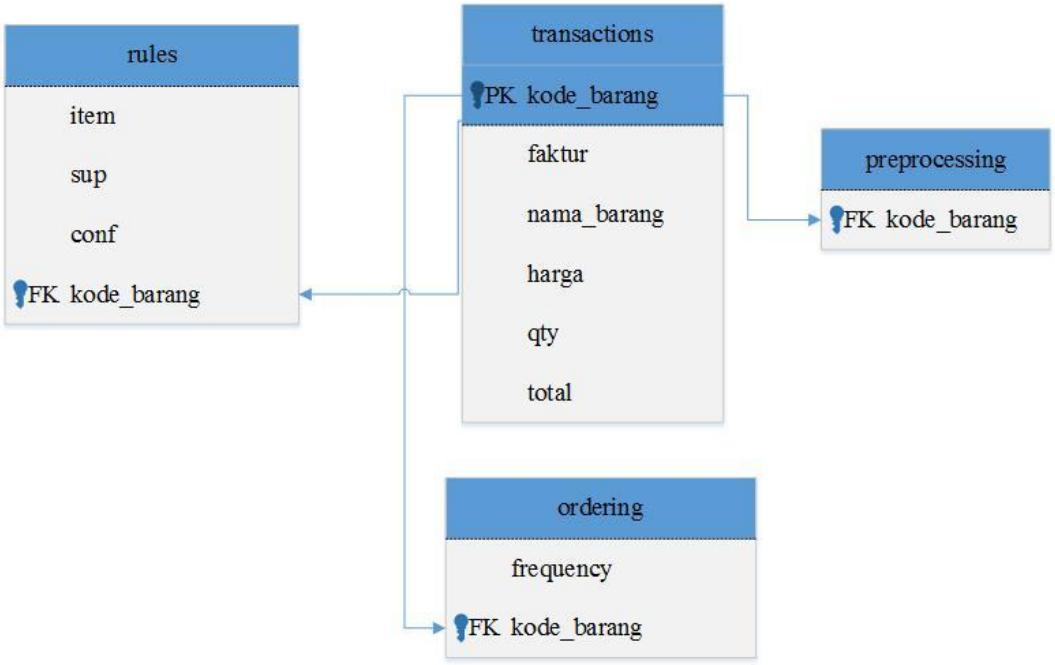
Nama *class* : BatasanForm Jenis *class* : *views*		
Nama	Jenis	*Type*
connStr: string	Atribut	Private
BatasanForm()	Method	Public
penetapanHarga()	Method	Private
buttonProsesBatasan_Click()	Method	Private

3.2.2 Perancangan Data

Perancangan data merupakan penjelasan mengenai data-data yang akan digunakan pada sistem. Perancangan data dibagi menjadi 2 yaitu skema relasi dan struktur tabel.

3.2.2.1 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 Primer Koperasi Kartika:



Gambar 3. 29 Skema relasi

3.2.2.2 Struktur Tabel

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

1. Tabel *transaction*

Tabel transactions adalah tabel yang digunakan untuk menyimpan data transaksi.

Tabel 3. 39 Tabel *transactions*

Nama Field	Tipe Data	Panjang	Keterangan
id	int	10	
faktur	varchar	50	
kode_barang	int	10	Primary Key
nama_barang	varchar	255	
harga	int	10	
qty	int	10	
total	int	10	

2. Tabel *preprocessing*

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

Tabel 3. 40 Tabel preprocessing

Nama Field	Tipe Data	Panjang	Keterangan
kode_barang	int	10	FOREIGN KEY (kode_barang) REFERENCES `transaction`(kode_barang)

3. Tabel *ordering*

Tabel ordering digunakan untuk menyimpan data barang yang dibeli.

Tabel 3. 41 Tabel *ordering*

Nama Field	Tipe Data	Panjang	Keterangan
kode_barang	int	10	FOREIGN KEY (kode_barang) REFERENCES `transaction`(kode_barang)
frequency	int	10	

4. Tabel *rules*

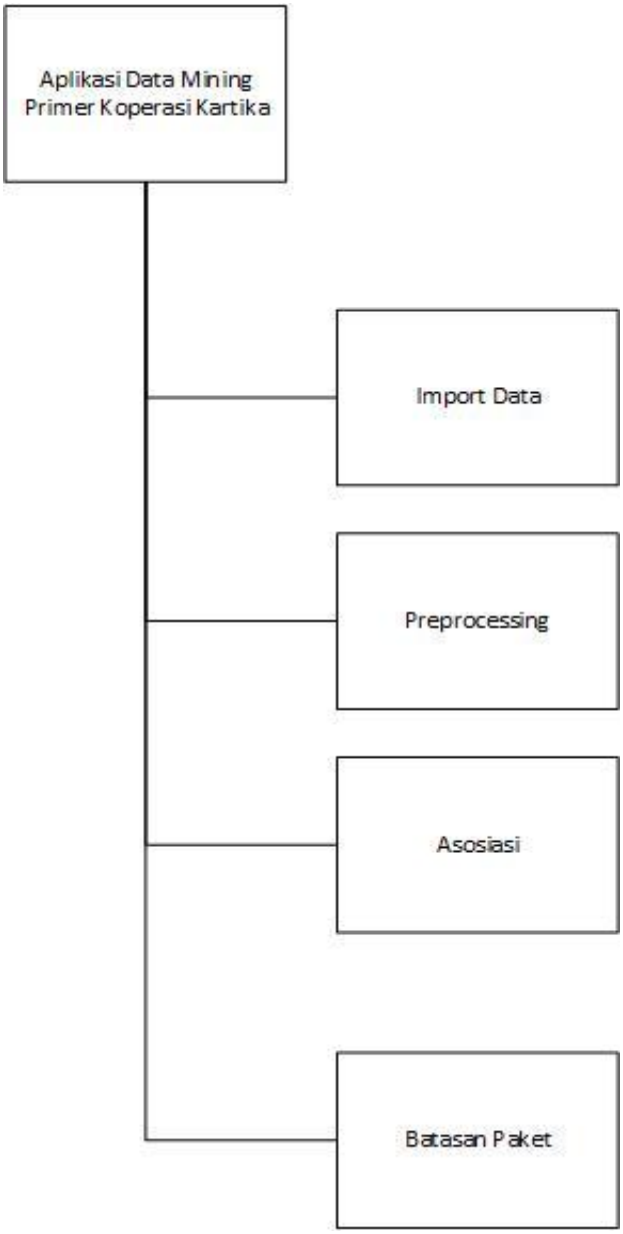
Tabel rules digunakan untuk menyimpan hasil rule

Tabel 3. 42 Tabel *rules*

Nama Field	Tipe Data	Panjang	Keterangan
Kode_barang	Int	10	FOREIGN KEY (kode_barang) REFERENCES `transaction`(kode_barang)
item	varchar	255	
sup	double		
conf	double		

3.2.3 Perancangan Struktur Menu

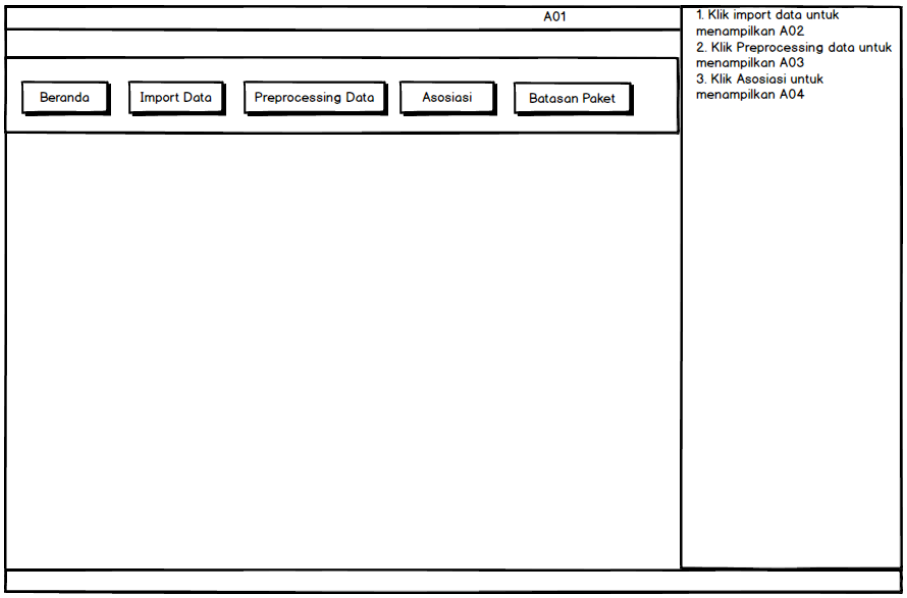
Dalam perancangan sebuah perangkat lunak dibutuhkan struktur menuyang berisikan menu dan submenu yang berfungsi untuk memudahkan user dalam menggunakan perangkat lunak tersebut. Berikut ini digambarkan mengenai struktur menu dalam aplikasi ini :



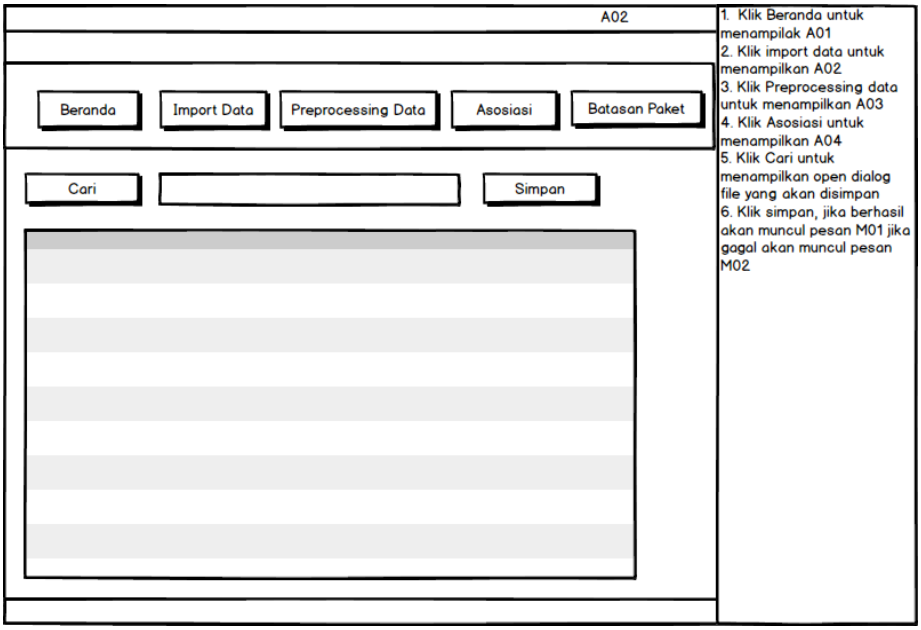
Gambar 3. 30 Struktur menu

3.2.4 Perancangan Antar Muka

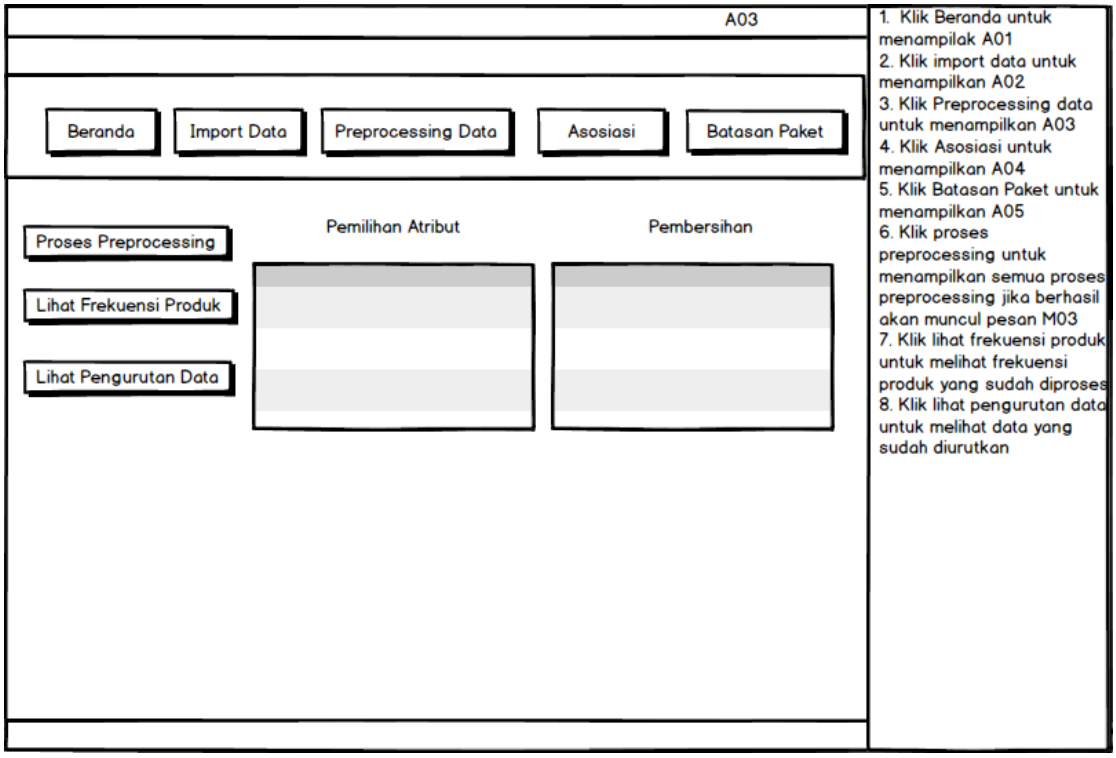
Perancangan antar muka berfungsi untuk memberi gambaran tentang pernagkat lunak yang akan dibangun, supaya lebih memudahkan dalam pembuatan perangkat lunak nantinya. Adapun perancangan antar muka untuk perangakt lunak yang akan dibangun akan dijelaskan berikut ini:



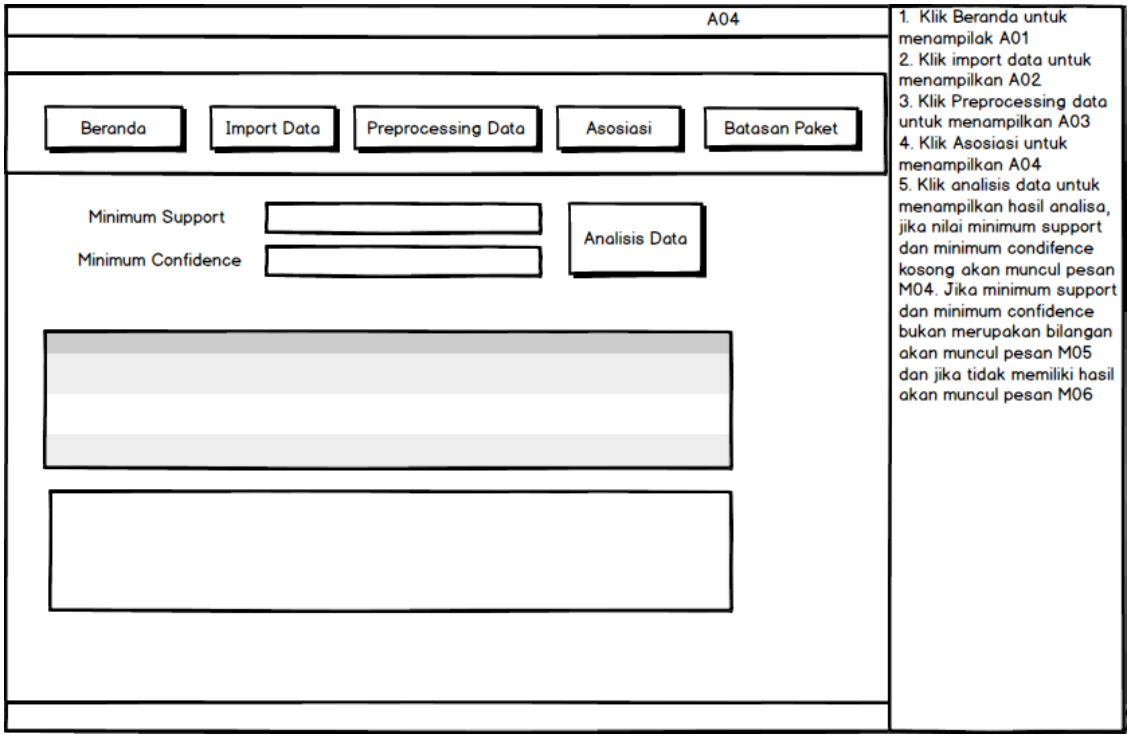
Gambar 3. 31 Perancangan antar muka menu utama



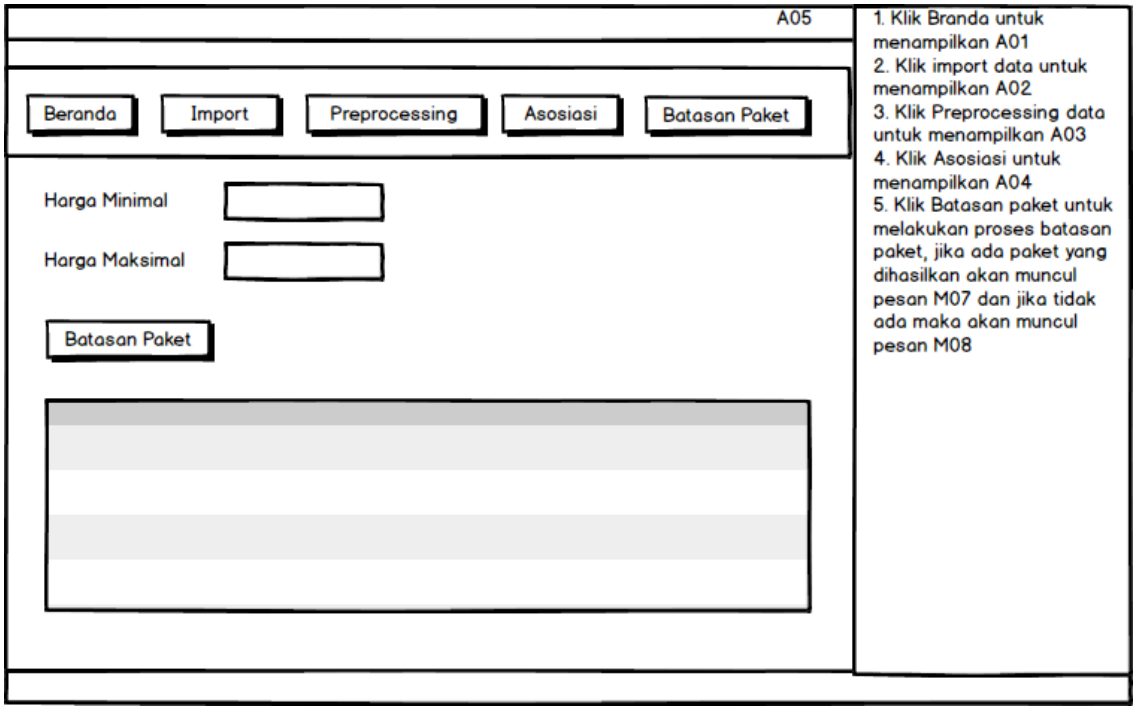
Gambar 3. 32 Perancangan antarmuka data *import*



Gambar 3. 33 Perancangan antarmuka *preprocessing*



Gambar 3. 34 Perancangan antar muka asosiasi



Gambar 3. 35 Perancangan antar muka batasan paket

3.2.5 Perancangan Pesan

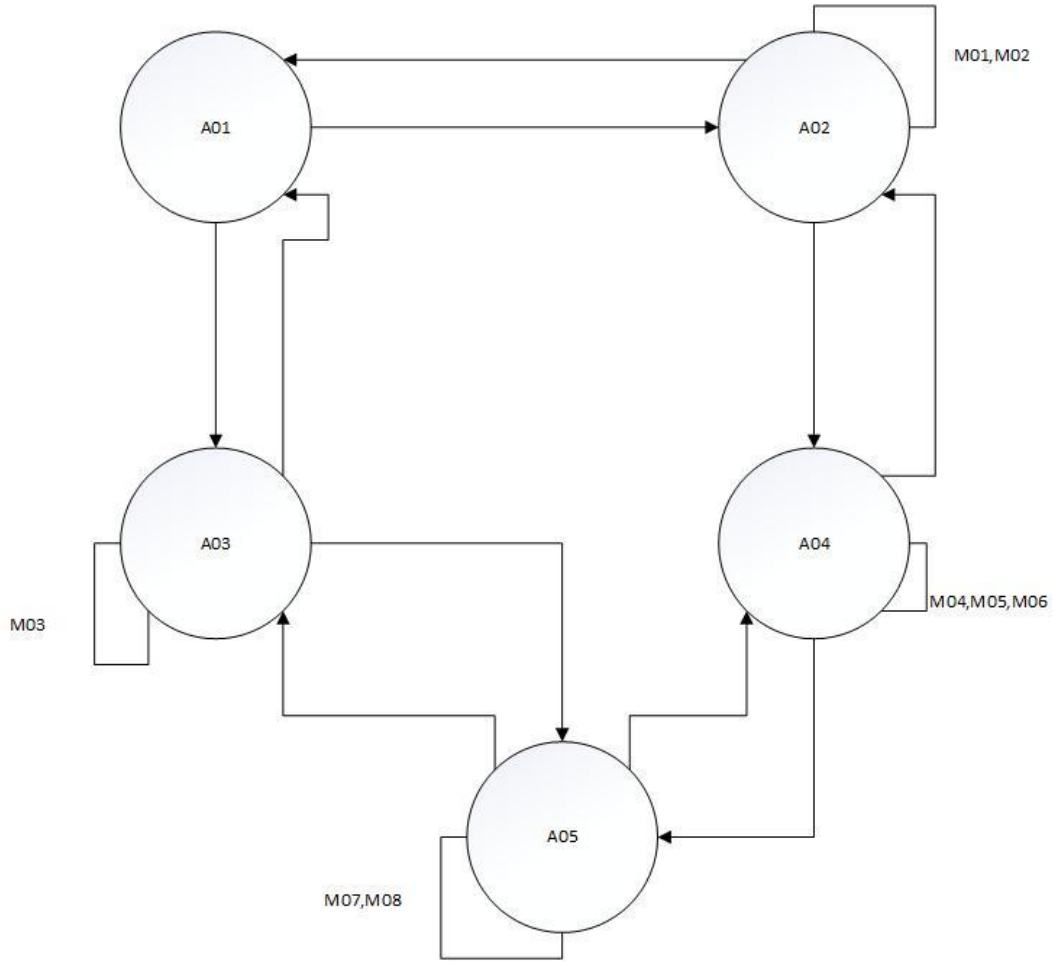
Adapun perancangan pesan dapat dilihat pada Tabel 3. 43:

Tabel 3. 43 Perancangan pesan

Kode Pesan	Pesan	No. Form
M01	Data Berhasil Disimpan	A02
M02	Data Gagal Disimpan	A02
M03	Preprocessing Data Berhasil Dilakukan	A03
M04	Minimum Support dan Minimum Confidence tidak boleh kosong	A04
M05	Rules tidak ditemukan	A04
M06	Paket berhasil terbentuk	A05

3.2.6 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 3. 36 berikut ini:



Gambar 3. 36 Jaringan Semantik