BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem (*System Analysis*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Dalam analisa sistem ini meluputi beberapa bagian, yaitu :

1. Analisis Masalah
2. Analisis *Crisp-DM*
3. Analisis *Non Fungsional* dan Kebutuhan *Non Fungsional*
4. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

Analisis masalah yang dilakukan adalah melihat dan mengidentifkasi permasalahan atau kendala di dalam penelitian yang dilakukan. Masalah dapat diartikan sebagai suatu pernyataan yang menyatakan tentang situasi yang memerlukan pemecahan. Perlu ada analysis masalah memalui rumusan masalah yang sudah ditentukan sebelumnya. Berdasarkan hasil wawancara dapat disimpulkan bahwa permasalahan yang ada di Mini Market Barokah yaitu kesulitan untuk menentukan barang yang akan dijadikan dalam satu paket.

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 Mini Market Barokah mempunyai tujuan bisnis yaitu untuk memenuhi permintaan konsumen untuk meningkatkan penjualan.

2. Menilai Situasi

   Menilai situasi merupakan tahapan untuk menjelaskan sumber daya yang tersedia di mini market barokah berikut merupakan sumber daya yang ada pada mini market barokah.

   a. Sumber Daya Perangkat keras
      1. Terdapat alat penunjang transaksi yaitu *printer thermal* dan *cashdrawer* (laci kasir)
      2. Komputer
      3. *Printer inject*
   b. Sumber Daya Data
      1. Sumber daya data yang di Mini Market Barokah meliputi data transaksi, data stok barang, data pegawai
      2. Sumber daya data di mini market barokah ini bersifat internal yaitu tersimpan di *database* mini market barokah
   c. Sumber Daya Personil
      1. Bagian pelayan yang bertugas melakukan transaksi pembelian, mencetak nota dan membuat laporan penjualan yang akan diserahkan ke pemilik
      2. Bagian *house keeping* yang bertanggung jawab dalam kebersihan di dalam atau diluar lingkungan mini market
      3. Bagian Admin Bertanggung jawab terhadap hal-hal yang menyangkut keuangan dan menghitung pemasukan dan pengeluaran setiap bulannya.

d. Kebutuhan

1. Data yang dipakai dalam penelitan ini adalah data transaksi

e. Asumsi

1. Data Transaksi yang digunakan merupakan hasil export dari aplikasi kasir

2. Di mini market barokah dengan format MS Excel(xlxs)

f. Batasan

1. Hasil dari *data mining* ini berupa rule sebagai acuan untuk membuat paket barang.
2. Hasil dari *generate rule* akan dilakukan *filtering rule*, yaitu memilih beberapa barang yang ada di dalam *rule* dan menjumlahkan harga dari barang yang dipilih oleh pengguna.
3. Batasan harga minimal dan maksimal akan ditentukan oleh pengguna.

3. Penentuan sasaran *Data mining*

Tujuan dari penerapan *data mining* pada pembelian konsumen ini adalah untuk mengetahui pola penentuan paket barang berdasarkan dari data yang sudah ada untuk meningkatkan penjualan.

3.1.2.2 Pemahaman Data

Tahapan pemahaman data dimulai dengan pengumpulan data awal dan hasil kegiatan dalam rangka untuk membiasakan diri dengan data untuk mengidentifikasi masalah data, untuk menentukan wawasan pertama ke data atau mendeteksi subset menarik untuk membentuk hipotesis untuk informasi yang tersembunyi.

1. Mengumpulkan Data Awal

Sumber data yang dalam penelitian ini yakni dari nota barang di mini market barokah. Berikut merupakan sumber data dalam bentuk nota yang dapat dilihat pada gambar 3.1 dibawah ini:


BAROKAH
Jl.raya darangdang No 140,Purwakarta
No Transaksi:16050017 Kasir Ahmad
Tanggal:16/05/2016
Sedap Goreng 3 1.500 4.500
Roma Kelapa 1 8.000 8.000
Total item 2 : 12.500
Disc % 0
Total: 12.500
TUNAI: 15.000
KEMBALIAN: 2.500

Gambar 3. 1 Data Transaksi Mini Market Barokah

Lalu data dari nota penjualan dipindahkan ke dalam *file Microsoft Excel* (.xlsx) seperti pada tabel 3.1 di bawah:

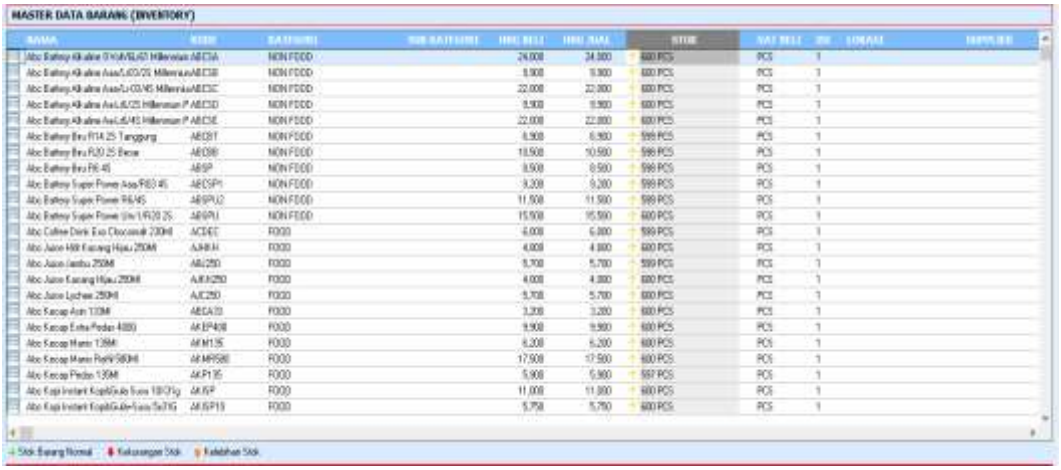
Tabel 3. 1 tabel data transaksi dari nota

No. Transaksi	Tanggal Transaksi	Nama Barang	QTY	Harga	Jumlah	Total Harga
15080001	1-Aug-15	MAGNUM BLACK	3 BKS	Rp 18,000	Rp54,000	Rp209,100
15080001	1-Aug-15	PEPSODENT 75GR	2 PCS	Rp 3,500	Rp7,000	Rp209,100
15080001	1-Aug-15	STARMILD	2 BKS	Rp 19,000	Rp38,000	Rp209,100
15080001	1-Aug-15	BERAS	1 KG	Rp 16,100	Rp16,100	Rp209,100
15080001	1-Aug-15	GOOD DAY MOCCA 3 IN 1	2 RCG	Rp 9,600	Rp19,200	Rp209,100
15080001	1-Aug-15	KOPI KAPAL API MIX	2 RCG	Rp 12,650	Rp25,300	Rp209,100
15080001	1-Aug-15	SEDAP GORENG	1 PCS	Rp 1,500	Rp1,500	Rp209,100
15080001	1-Aug-15	TELUR AYAM	3 KG	Rp 16,000	Rp48,000	Rp209,100
15080002	1-Aug-15	CLASMILD 16	2 BKS	Rp 14,000	Rp28,000	Rp254,450
15080002	1-Aug-15	DJI SAM SOE	2 BKS	Rp 18,000	Rp36,000	Rp254,450

No. Transaksi	Tanggal Transaksi	Nama Barang	QTY	Harga	Jumlah	Total Harga
15080002	1-Aug-15	JARUM SUPER 12	2 BKS	Rp 15,000	Rp30,000	Rp254,450
15080002	1-Aug-15	LIFEBUOY MERAH	1 PCS	Rp 2,500	Rp2,500	Rp254,450
15080002	1-Aug-15	BERAS	3 KG	Rp 16,100	Rp48,300	Rp254,450
15080002	1-Aug-15	GOOD DAY MOCCA 3 IN 1	3 RCG	Rp 9,600	Rp28,800	Rp254,450
15080002	1-Aug-15	GULA MERAH	1 KG	Rp 3,500	Rp3,500	Rp254,450
15080002	1-Aug-15	INDOCAFE MIX	2 RCG	Rp 4,800	Rp9,600	Rp254,450
15080002	1-Aug-15	KOPI KAPAL API MIX	5 RCG	Rp 12,650	Rp63,250	Rp254,450
15080002	1-Aug-15	SEDAP GORENG	3 PCS	Rp 1,500	Rp4,500	Rp254,450

Untuk pemaketan sendiri, dibutuhkan kode dan jenis untuk melihat apa yang akan dibutuhkan dalam tahap selanjutnya. Untuk kode barang dan jenis dapat dilihat dari tabel stok barang, contoh dar tabel stok barang dapat dilihat pada tabel 3.2 di bawah:

Tabel 3. 2 contoh data stok barang



Selanjutnya, kode barang dan kategori dimasukan dan disatukan ke dalam *file* data transaksi yang telah tersedia, berikut hasil setelah kode barang dan kategori yang telah dimasukan dapat dilihat pada tabel 3.3 dapat dilihat pada tabel di bawah:

Tabel 3. 3 Data Transaksi

No. Transaksi	Tanggal Transaksi	Kode Barang	Nama Barang	Kategori	QTY	Harga	Jumlah	Total Harga
15080001	1-Aug-15	RMB03	MAGNUM BLACK	NON FOO	3 BKS	Rp 18,000	Rp54,000	Rp209,100
15080001	1-Aug-15	PPS99	PEPSODENT 75GR	NON FOO	2 PCS	Rp 3,500	Rp7,000	Rp209,100
15080001	1-Aug-15	RSTM1	STARMILD	NON FOO	2 BKS	Rp 19,000	Rp38,000	Rp209,100
15080001	1-Aug-15	BRS01	BERAS	SEMBAKO	1 KG	Rp 16,100	Rp16,100	Rp209,100
15080001	1-Aug-15	GDM11	GOOD DAY MOCCA 3 IN 1	FOOD	2 RCG	Rp 9,600	Rp19,200	Rp209,100
15080001	1-Aug-15	KPAM1	KOPI KAPAL API MIX	FOOD	2 RCG	Rp 12,650	Rp25,300	Rp209,100
15080001	1-Aug-15	SDG51	SEDAP GORENG	FOOD	1 PCS	Rp 1,500	Rp1,500	Rp209,100
15080001	1-Aug-15	STM21	TELUR AYAM	SEMBAKO	3 KG	Rp 16,000	Rp48,000	Rp209,100
15080002	1-Aug-15	RCM04	CLASMILD 16	NON FOO	2 BKS	Rp 14,000	Rp28,000	Rp254,450
15080002	1-Aug-15	RDS02	DJI SAM SOE	NON FOO	2 BKS	Rp 18,000	Rp36,000	Rp254,450
15080002	1-Aug-15	RJS12	JARUM SUPER 12	NON FOO	2 BKS	Rp 15,000	Rp30,000	Rp254,450
15080002	1-Aug-15	SLM02	LIFEBUOY MERAH	NON FOO	1 PCS	Rp 2,500	Rp2,500	Rp254,450
15080002	1-Aug-15	BRS01	BERAS	SEMBAKO	3 KG	Rp 16,100	Rp48,300	Rp254,450
15080002	1-Aug-15	GDM11	GOOD DAY MOCCA 3 IN 1	FOOD	3 RCG	Rp 9,600	Rp28,800	Rp254,450
15080002	1-Aug-15	SGM11	GULA MERAH	SEMBAKO	1 KG	Rp 3,500	Rp3,500	Rp254,450
15080002	1-Aug-15	INM03	INDOCAFE MIX	FOOD	2 RCG	Rp 4,800	Rp9,600	Rp254,450
15080002	1-Aug-15	KPAM1	KOPI KAPAL API MIX	FOOD	5 RCG	Rp 12,650	Rp63,250	Rp254,450
15080002	1-Aug-15	SDG51	SEDAP GORENG	FOOD	3 PCS	Rp 1,500	Rp4,500	Rp254,450

2. Mendeskripsikan Data

   Adapun detail informasi mengenai data transaksi pemesanan yang digunakan dapat dilihat pada tabel 3.4 dibawah ini :

Tabel 3. 4 Struktur Data Transaksi

Keterangan		
Deskripsi	Data ini berisi menegenai data transaksi di mini market barokah	
Format	Microsoft Excel (.xlsx)	
Keterangan		
Atribut	No Transaksi	Nomor pembelian
	Tanggal Transaksi	Tanggal proses pembelian
	Kode Barang	Kode dari setiap barang
	Nama Barang	Nama barang yang dibeli
	Kategori	Kategori dari setiap nama barang
	Qty	Quantity
	Harga	Harga barang yang dibeli
	Jumlah	Jumlah Harga barang yang dibeli
	Total Harga	Total harga yang dibeli

3. Kualitas Data

Kualitas data merupakan tahap dimana dlakukan pengecekan apakah data tersebut memiliki noise atau tidak. Hasil dari verifikasi kualitas data yang digunakan dalam penelitian ini tidak terdapat noise dari 26 transaksi. Yang dimaksud dengan noise disini adalah 1 transaksi yang hanya membeli 1 barang . Maka dari itu data – data yang memiliki *noise* akan dilakukan pembersihan data pada tahap data *preparation*.

3.1.2.3 Persiapan Data

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

1. Ekstraksi Data

Ekstraksi data dapat diartikan sebagai proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data ketingkat selanjutnya ataupun untuk menyimpan data hasil ekstrak tersebut. Dalam penelitian ini, data yang berasal dari *flat file* berformat *microsoft excel* di-ekstrak, kemudian disimpan kedalam sebuah *database* agar memudahkan dalam proses pengolahan data. Berikut merupakan contoh sampel data transaksi bulan Agustus 2015 yang diambil sebanyak 26 transaksi sebanyak 198 *record* yang terdapat dalam data tersebut adalah No Transaksi, Tanggal Transaksi, Kode Barang, Nama Barang, Kategori, qty, Harga, Jumlah, Total Harga. Data transaksi dapat dilihat pada table 3.4 di bawah ini:.

Tabel 3. 5 contoh data transaksi

No. Transaksi	Tanggal Transaksi	Kode Barang	Nama Barang	Kategori	QTY	Harga	Jumlah	Total Harga
15080001	1-Aug-15	RMB03	MAGNUM BLACK	NON FOOD	3 BKS	Rp 18,000	Rp54,000	Rp209,100
15080001	1-Aug-15	PPS99	PEPSODENT 75GR	NON FOOD	2 PCS	Rp 3,500	Rp7,000	Rp209,100

No. Transaksi	Tanggal Transaksi	Kode Barang	Nama Barang	Kategori	QTY	Harga	Jumlah	Total Harga
15080001	1-Aug-15	RSTM1	STARMILD	NON FOOD	2 BKS	Rp 19,000	Rp38,000	Rp209,100
15080001	1-Aug-15	BRS01	BERAS	SEMBAKO	1 KG	Rp 16,100	Rp16,100	Rp209,100

2. Pembersihan Data

Proses menghilangkan noise dan menghilangkan data tidak relevan atau inkonsisten disebut pembersihan data. Dalam penelitian ini yaitu menghilangkan record pada data transaksi yang memiliki kategori *non food* dan menghilangkan transaksi yang memiliki barang tunggal. Hasil pembersihan data dapat dilihat pada tabel 3.5 dan 3.6 di bawah ini:

Tabel 3. 6 sebelum pembersihan

No. Transaksi	Tanggal Transaksi	Kode Barang	Nama Barang	Kategori
15080001	1-Aug-15	RMB03	MAGNUM BLACK	NON FOOD
15080001	1-Aug-15	PPS99	PEPSODENT 75GR	NON FOOD
15080001	1-Aug-15	RSTM1	STARMILD	NON FOOD
15080001	1-Aug-15	BRS01	BERAS	SEMBAKO
15080001	1-Aug-15	GDM11	GOOD DAY MOCCA 3 IN 1	FOOD
15080001	1-Aug-15	KPAM1	KOPI KAPAL API MIX	FOOD
15080001	1-Aug-15	SDG51	SEDAP GORENG	FOOD

Tabel 3. 7 sesudah pembersihan

No. Transaksi	Tanggal Transaksi	Kode Barang	Nama Barang	Kategori
15080001	1-Aug-15	BRS01	BERAS	SEMBAKO
15080001	1-Aug-15	GDM11	GOOD DAY MOCCA 3 IN 1	FOOD
15080001	1-Aug-15	KPAM1	KOPI KAPAL API MIX	FOOD
15080001	1-Aug-15	SDG51	SEDAP GORENG	FOOD

3. Pemilihan Atribut

Setelah proses pembersihan data kemudian dilakukan proses pemilihan atribut, proses dimana atribut data akan dipilih dan diproses sesuai dengan kebutuhan *data mining*. Dalam penelitian ini, atribut yang dibutuhkan hanya 2 atribut yaitu No Transaksi dan Nama Barang. Kedua atribut ini digunakan untuk memenuhi tujuan awal dimana akan dicari pola pembelian konsumen berdasarkan barang yang dibeli. Seperti atribut No

Transaksi digunakan untuk membedakan satu transaksi dengan transakaksi lainya, dan atribut Nama Barang digunakan untuk mengetahui barang apa saja yang dibeli dalam satu transaksi. Hasil pemilihan atribut dapat dilihat pada tabel 3.8 di bawah ini:

Tabel 3. 8 Pemilihan Atribut

No. Transaksi	Nama Barang
15080001	GOOD DAY MOCCA 3 IN 1
15080001	KOPI KAPAL API MIX
15080001	SEDAP GORENG
15080001	BERAS
15080001	TELUR AYAM

3.1.2.4 Modeling

Penelitian ini bertujuan untuk mendapatkan informasi mengenai pola pembelian konsumen yang nantinya akan digunakan oleh pihak Mini Market Barokah sebagai dasar pengambilan keputusan untuk menentukan barang apa saja yang akan ditawarkan ke konsumen.

Dengan menerapkan *Data mining*, data transaksi akan diolah dengan aturan asosiasi atau *Association Rules* untuk menemukan pola-pola pembelian barang yang sering dibeli oleh konsumen. *Association Rules* terbagi menjadi dua tahap yaitu:

1. Mencari *frequent itemset*, pada tahap ini akan menggunakan algoritma *CT-Pro* dengan langkah sebagai berikut:
   a. Menghitung jumlah kemunculan tiap barang
   b. Seleksi dengan menentukan nilai *minimum support*
   c. Membangun *Id Global Item*
2. *Generate Rule*

1. Mencari *Frequent Itemset*

   a. Menghitung Jumlah Kemunculan Tiap Barang

   Dengan menggunakan tabel D-3 pada lampiran D, masing-masing barang dihitung jumlah kemunculan. Hasil dari penghitungan kemunculan barang dapat dilihat pada tabel 3.9 dibawah ini:

Tabel 3. 9 Hasil perhitungan kemunculan barang

Nama Barang	*Support Count*
SEDAP GORENG	15
BERAS	15
TELUR AYAM	15
KOPI KAPAL API MIX	14
INDOCAFE MIX	13
GOOD DAY MOCCA 3 IN 1	9
ABC SUSU KOPI & GULA	8
KECAP BANGO 220ML	8
GULA PASIR	6
SARIMI KUAH	5
ROMA KELAPA	5
MARI SUSU	3
BENG-BENG	2
GULA MERAH	2
ES TELER	2
EXTRA JOSS	1
ROTI ANISA 1000	1
KUE 4000	1
BETTER	1
SUKRO	1

   b. Seleksi Dengan Menentukan Nilai *Minimum Support*

   Setelah dihitung jumlah kemunculan masing-masing barang maka selanjutnya akan ditentukan nilai *minimum support* untuk melihat batasan terendah munculnya barang.

   Dalam menentukan nilai *minimum support* dapat berubah sesuai kebutuhan *user*. Hal ini juga diperkuat dengan pernyatan pakar *data mining* Daniel T. Larose, yang menerangkan bahwa *user* bebas menentukan nilai *minimum support* (*minsup*) dan *minimum confidence* (*mincof*) sesuai

kebutuhan. Karena, jika nilai *minsup* terlalu rendah maka akan mengakibatkan *rule* asosiasi yang dihasilkan terlalu banyak dan akan ada *rule* yang kurang bermakna. Sebaliknya jika nilai *minimum support* terlalu tinggi maka akan mengakibatkan *rule* asosiasi terlalu sedikit atau bahkan tidak dapat dihasilkan sama sekali.

Maka dalam kasus ini batasan barang yang muncul harus sebanyak >= 5 kali. Berdasarkan tabel 3.9 untuk barang yang *support count*-nya tidak memenuhi batasan *minimum support* dieliminasi. Berikut adalah tabel hasil eliminasi kemunculan barang yang tidak memenuhi *minimun suport* yang diacu pada tabel 3.10 di bawah ini:

Tabel 3. 10 Hasil elimnasi kemunculan barang

Nama Barang	*Support Count*
SEDAP GORENG	15
BERAS	15
TELUR AYAM	15
KOPI KAPAL API MIX	14
INDOCAFE MIX	13
GOOD DAY MOCCA 3 IN 1	9
ABC SUSU KOPI & GULA	8
KECAP BANGO 220ML	8
GULA PASIR	6
SARIMI KUAH	5
ROMA KELAPA	5

Begitupula pada data transaksi yang ada pada tabel D-3 pada lampiran D, untuk barang yang *support count*-nya tidak memenuhi batasan minimum *support* juga dieliminasi dan hasilnya dapat dilihat pada tabel 3.11:

Tabel 3. 11 Data Transaksi yang *frequent*

No. Transaksi	Nama Barang
15080001	BERAS
15080001	GOOD DAY MOCCA 3 IN 1
15080001	KOPI KAPAL API MIX
15080001	SEDAP GORENG
15080001	TELUR AYAM
15080002	BERAS
15080002	GOOD DAY MOCCA 3 IN 1
15080002	INDOCAFE MIX
15080002	KOPI KAPAL API MIX
15080002	SEDAP GORENG
15080003	SEDAP GORENG
5080003	TELUR AYAM
15080004	ABC SUSU KOPI & GULA
15080004	BERAS
15080004	GOOD DAY MOCCA 3 IN 1
15080004	INDOCAFE MIX
15080004	KOPI KAPAL API MIX
15080004	SEDAP GORENG
15080004	TELUR AYAM
15080005	INDOCAFE MIX
15080005	ROMA KELAPA
15080005	TELUR AYAM
15080006	ABC SUSU KOPI & GULA
15080006	KOPI KAPAL API MIX
15080006	SEDAP GORENG
15080007	ABC SUSU KOPI & GULA
15080007	ROMA KELAPA
15080007	SARIMI KUAH
15080007	TELUR AYAM
15080008	BERAS
15080008	GOOD DAY MOCCA 3 IN 1
15080008	GULA PASIR
15080008	INDOCAFE MIX
15080008	KECAP BANGO 220ML
15080008	KOPI KAPAL API MIX
15080008	SARIMI KUAH
15080008	SEDAP GORENG
15080009	GOOD DAY MOCCA 3 IN 1

No. Transaksi	Nama Barang
15080015	BERAS
15080015	GOOD DAY MOCCA 3 IN 1
15080015	GULA PASIR
15080015	INDOCAFE MIX
15080015	KECAP BANGO 220ML
15080015	KOPI KAPAL API MIX
15080015	TELUR AYAM
15080016	GOOD DAY MOCCA 3 IN 1
15080016	SEDAP GORENG
15080017	BERAS
15080017	KOPI KAPAL API MIX
15080018	BERAS
15080018	INDOCAFE MIX
15080018	SARIMI KUAH
15080018	TELUR AYAM
15080019	ABC SUSU KOPI & GULA
15080019	BERAS
15080019	KECAP BANGO 220ML
15080019	KOPI KAPAL API MIX
15080019	ROMA KELAPA
15080020	BERAS
15080020	INDOCAFE MIX
15080020	ROMA KELAPA
15080020	SEDAP GORENG
15080020	TELUR AYAM
15080021	BERAS
15080021	GOOD DAY MOCCA 3 IN 1
15080021	GULA PASIR
15080021	INDOCAFE MIX
15080021	KECAP BANGO 220ML
15080021	TELUR AYAM
15080022	KOPI KAPAL API MIX
15080022	SEDAP GORENG
15080023	ABC SUSU KOPI & GULA
15080023	BERAS
15080023	GULA PASIR
15080023	KECAP BANGO 220ML
15080023	TELUR AYAM

No. Transaksi	Nama Barang
15080009	TELUR AYAM
15080010	BERAS
15080010	INDOCAFE MIX
15080010	KECAP BANGO 220ML
15080010	KOPI KAPAL API MIX
15080010	SARIMI KUAH
15080010	SEDAP GORENG
15080011	BERAS
15080011	KOPI KAPAL API MIX
15080011	SEDAP GORENG
15080011	TELUR AYAM
15080012	INDOCAFE MIX
15080012	KOPI KAPAL API MIX
15080012	SEDAP GORENG
15080013	KECAP BANGO 220ML
15080013	TELUR AYAM
15080014	BERAS
15080014	ROMA KELAPA
15080014	SEDAP GORENG

No. Transaksi	Nama Barang
15080024	ABC SUSU KOPI & GULA
15080024	GULA PASIR
15080024	INDOCAFE MIX
15080024	KOPI KAPAL API MIX
15080024	SARIMI KUAH
15080024	SEDAP GORENG
15080024	TELUR AYAM
15080025	ABC SUSU KOPI & GULA
15080025	BERAS
15080025	GOOD DAY MOCCA 3 IN 1
15080025	GULA PASIR
15080025	INDOCAFE MIX
15080025	KECAP BANGO 220ML
15080025	KOPI KAPAL API MIX
15080025	SEDAP GORENG
15080026	ABC SUSU KOPI & GULA
15080026	INDOCAFE MIX
15080026	TELUR AYAM

c. Membangun *Id Global item*

Setelah proses pengeliminasian barang dilakukan, kemudian dilakukan pemberian *id global item* untuk memulai proses pembentukan *tree*. Pada tabel 3.5, barang yang sudah terurut dan sudah memenuhi *minimum support* akan dijadikan sebagai urutan kemunculan barang pada setiap transaksi yang mengacu pada *id global item*. Berikut adalah pemberian *id global item* yang diacu pada tabel 3.12 di bawah ini:

Tabel 3. 12 Pemberian *Id Global Item*

Nama Barang	*Support Count*	*Id Global Item*
SEDAP GORENG	15	1
BERAS	15	2
TELUR AYAM	15	3
KOPI KAPAL API MIX	14	4
INDOCAFE MIX	13	5

Nama Barang	*Support Count*	*Id Global Item*
GOOD DAY MOCCA 3 IN 1	9	6
ABC SUSU KOPI & GULA	8	7
KECAP BANGO 220ML	8	8
GULA PASIR	6	9
SARIMI KUAH	5	10
ROMA KELAPA	5	11

d. *Mapping* data

Mapping yaitu memetakan data transaksi dari table 3.11 terhadap tabel *id global item* pada table 3.12, dimana nama barang pada tabel 3.10 digantikan oleh *id* sesuai pada table 3.12.Data transaksi kemudian diurutkan dari id terkecil ke terbesar dari setia no transaksi. Berikut adalah *mapping* data transaksi pada tabel 3.11 yang dapat dilihat pada tabel 3.13 dibawah ini:

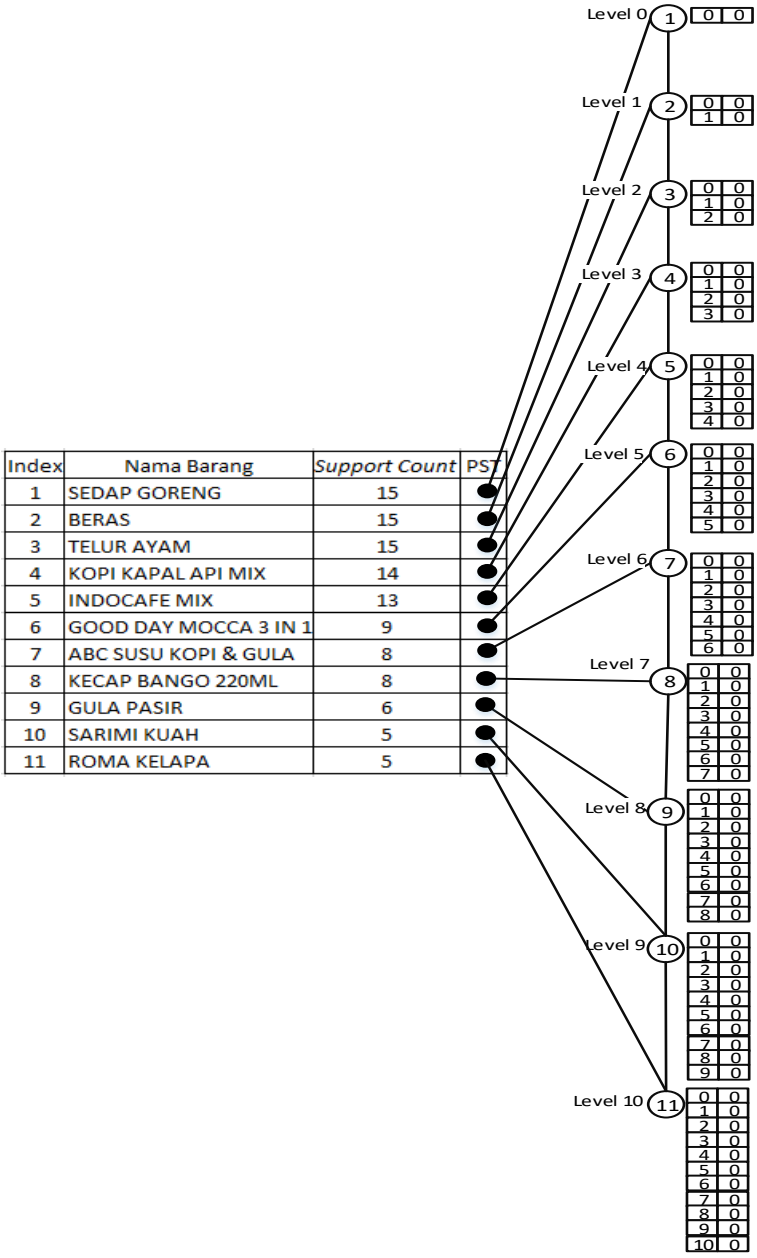
Tabel 3. 13 *Mapping*

No Transaksi	Nama Barang
15080001	1,2,3,4,6
15080002	1,2,4,5,6
15080003	1,3
15080004	1,2,3,4,5,6,7
15080005	3,5,11
15080006	1,4,7
15080007	3,7,10,11
15080008	1,2,4,5,6,8,9,10
15080009	3,6
15080010	1,2,4,5,8,10
15080011	1,2,3,4
15080012	1,4,5
15080013	3,8
15080014	1,2,11
15080015	2,3,4,5,6,8,9
15080016	1,6
15080017	2,4
15080018	2,3,5,10
15080019	2,4,7,8,11
15080020	1,2,3,5,11

No Transaksi	Nama Barang
15080021	2,3,5,6,8,9
15080022	1,4
15080023	2,3,7,8,9
15080024	1,3,4,5,7,9,10
15080025	1,2,4,5,6,7,8,9
15080026	3,5,7

e. Membuat *CFP-Tree*

Setelah proses mapping kemudian proses pembentukan *CFP-Tree*. Pada pembentukan *Global CFP-Tree* dimulai dengan membuat *left most branch* yang dapat dilihat pada gambar 3.2 pada saat proses pembentukan *left most branch* dibuat *node-node* yang berurutan sesuai dengan *global item table*. Setiap *node* memiliki level yang berbeda sesuai dengan tingkatannya. Tingkatan ini dibuat berdasarkan *global item* pada *table* 3.12



Index	Nama Barang	*Support Count*	PST
1	SEDAP GORENG	15	
2	BERAS	15	
3	TELUR AYAM	15	
4	KOPI KAPAL API MIX	14	
5	INDOCAFE MIX	13	
6	GOOD DAY MOCCA 3 IN 1	9	
7	ABC SUSU KOPI & GULA	8	
8	KECAP BANGO 220ML	8	
9	GULA PASIR	6	
10	SARIMI KUAH	5	
11	ROMA KELAPA	5	

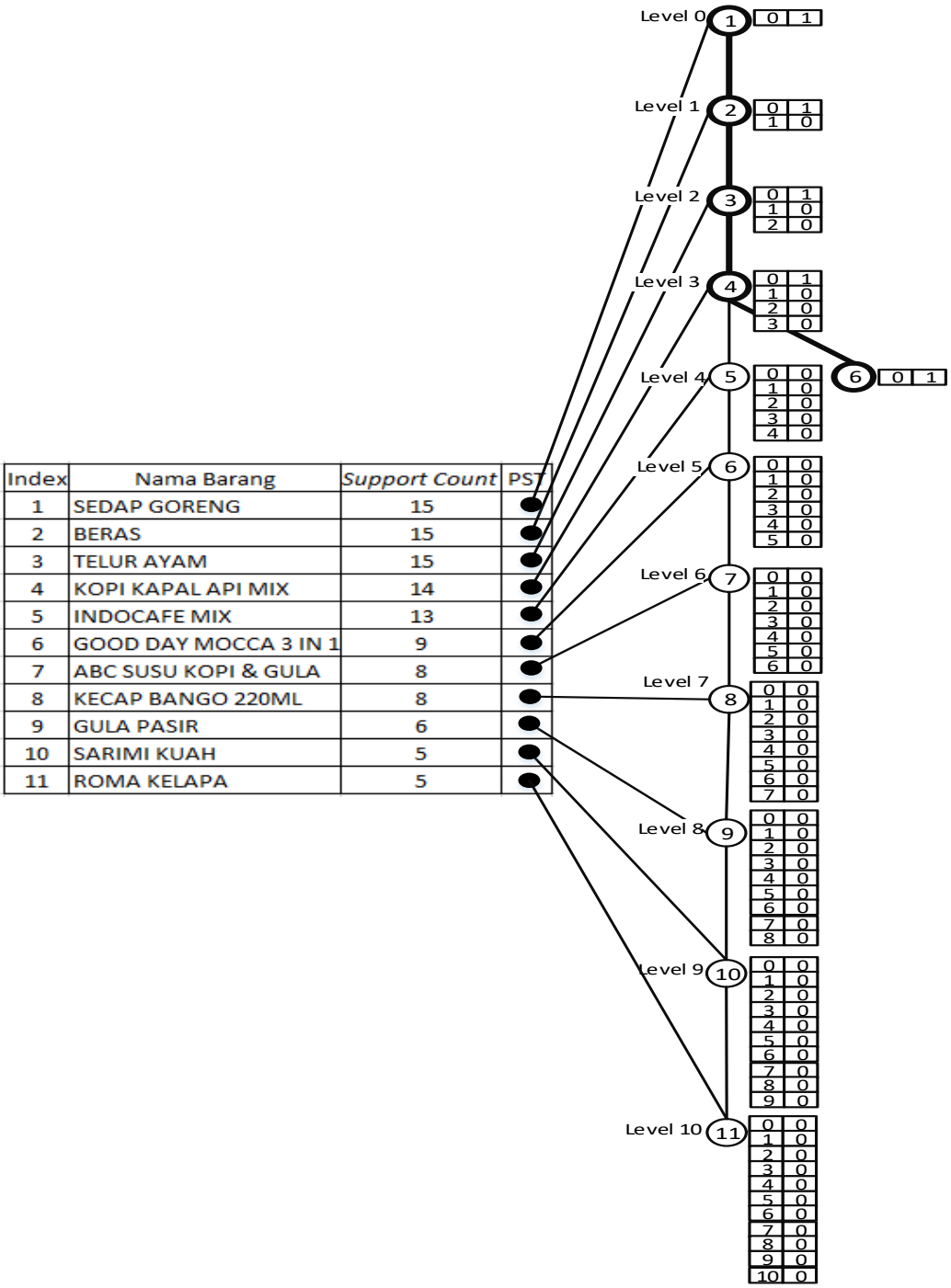
.

Gambar 3. 2 *left most branch*

Setelah pembentukan *Left most branch* selesai maka tahapan selanjutnya adalah *insert to CFP-Tree* yang memasukan data transaksi kedalam *tree*. Berikut adalah tahapannya:

1. Dimulai dari *item* yang pertama kali muncul dari no transaksi yang telah di *mapping*, *item* tersebut akan menjadi *curnode*
2. Tambahkan *count* = 1 pada item pertama muncul sesuai dengan levelnya.
3. Jika *currnode* sudah memiliki anak, dan *item* yang muncul sama dengan anak tersebut maka tambahkan *count* = 1 pada *level* yang sama *currnode*
4. Jika sudah memiliki anak, tetapi data yang muncul berbeda dengan anak dari *cuurnode*, buat *node* baru dan tambahkan *count* = 1 pada *level* yang sama *currnode*, hubungkan *node* baru tersebut pada *currnode* melalui *link node*

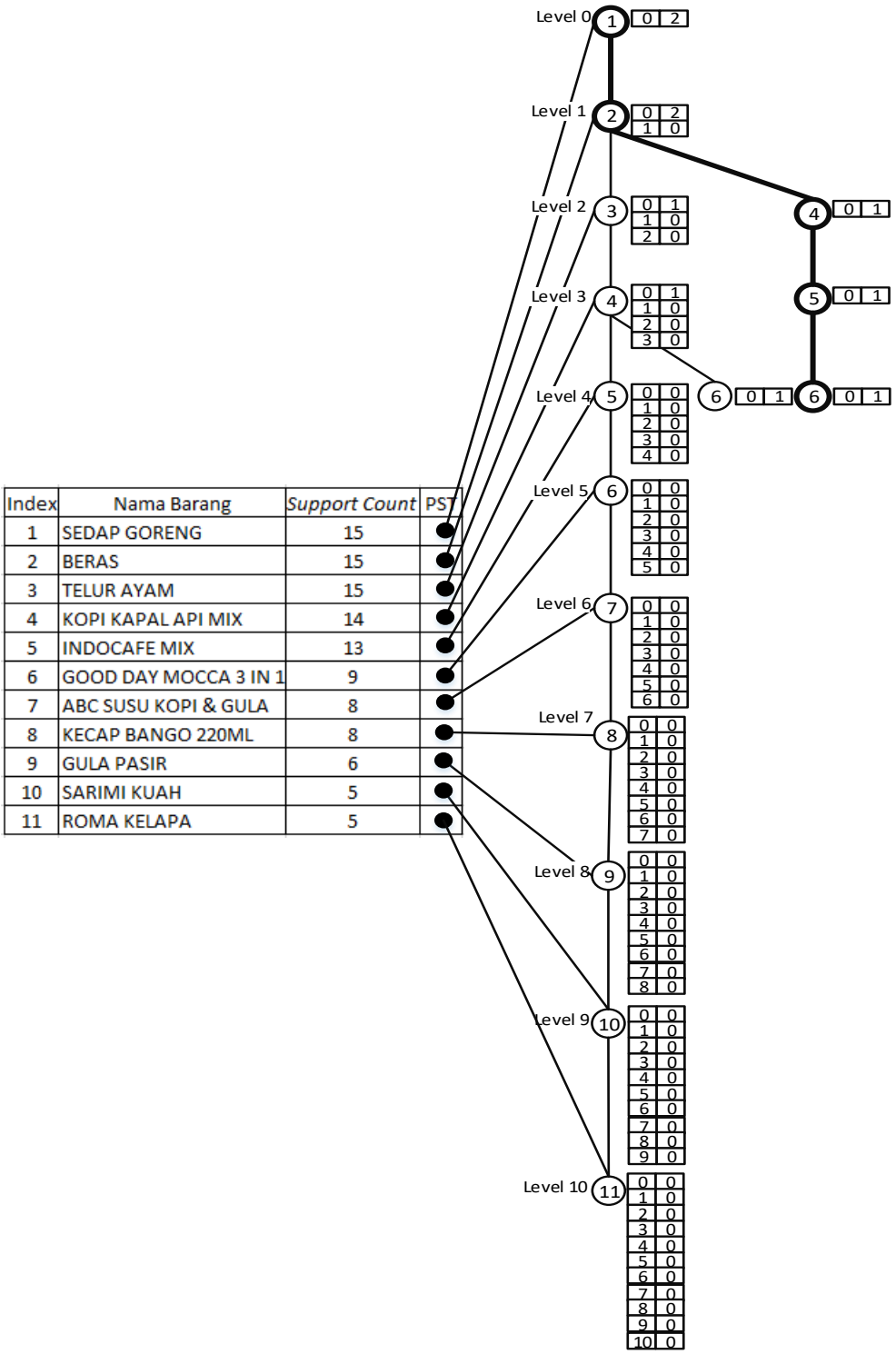
1. Nomor Transaksi 15080001 dengan *index* : 1,2, 3, 4, 6 pembentukan *CFP-Tree* dengan nomor transaksi 15080001 dapat dilihat pada gambar 3.4 sebagai berikut :



Index	Nama Barang	*Support Count*	PST
1	SEDAP GORENG	15	
2	BERAS	15	
3	TELUR AYAM	15	
4	KOPI KAPAL API MIX	14	
5	INDOCAFE MIX	13	
6	GOOD DAY MOCCA 3 IN 1	9	
7	ABC SUSU KOPI & GULA	8	
8	KECAP BANGO 220ML	8	
9	GULA PASIR	6	
10	SARIMI KUAH	5	
11	ROMA KELAPA	5	

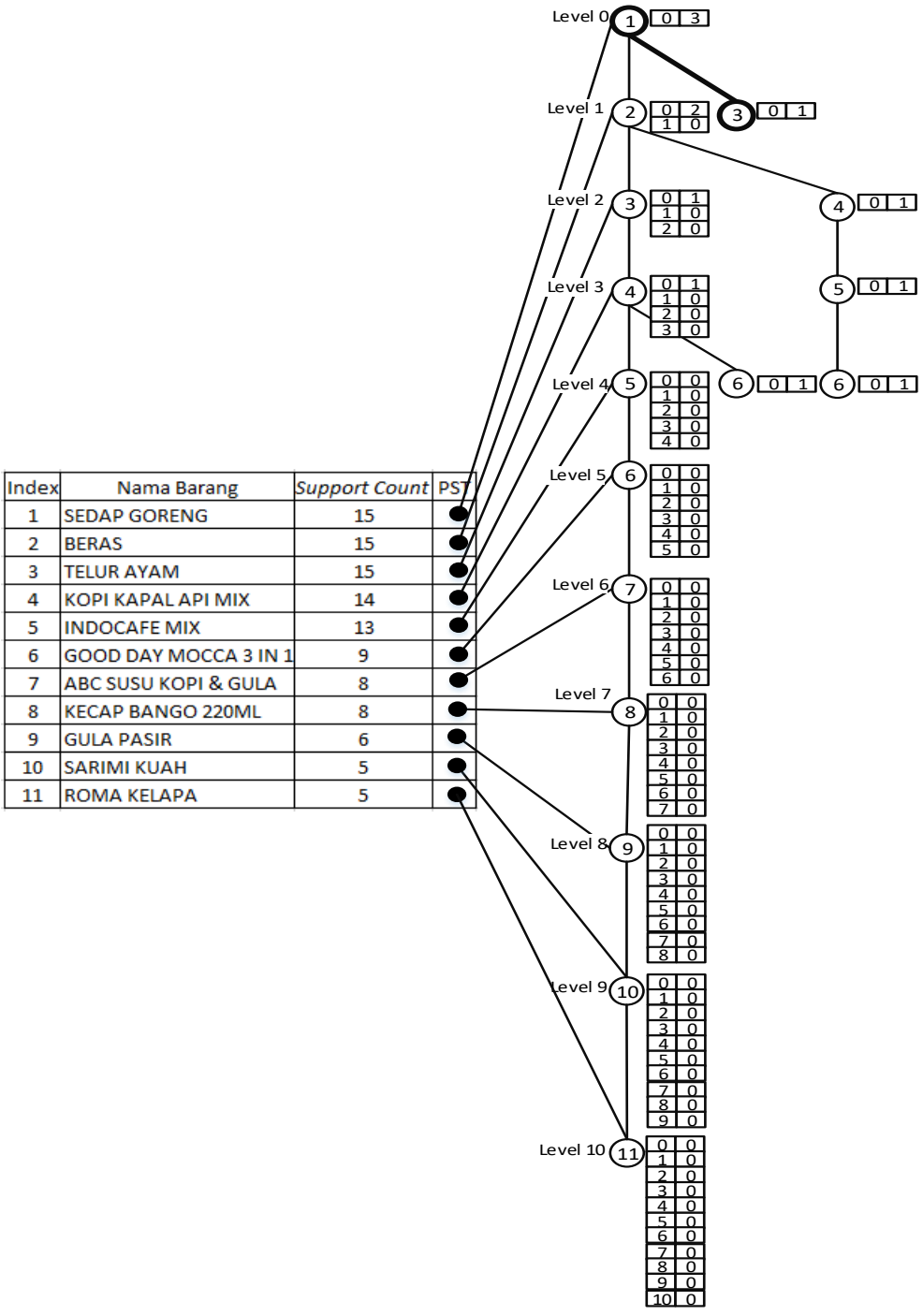
Gambar 3. 3 *CFP-Tree* Transaksi 15080001

2. Nomor Transaksi 15080002 dengan *index* 1, 2, 4, 5, 6 pembentukan *CFP-Tree* dengan nomor transaksi 15080002 dapat dilihat pada gambar 3.4 sebagai berikut :



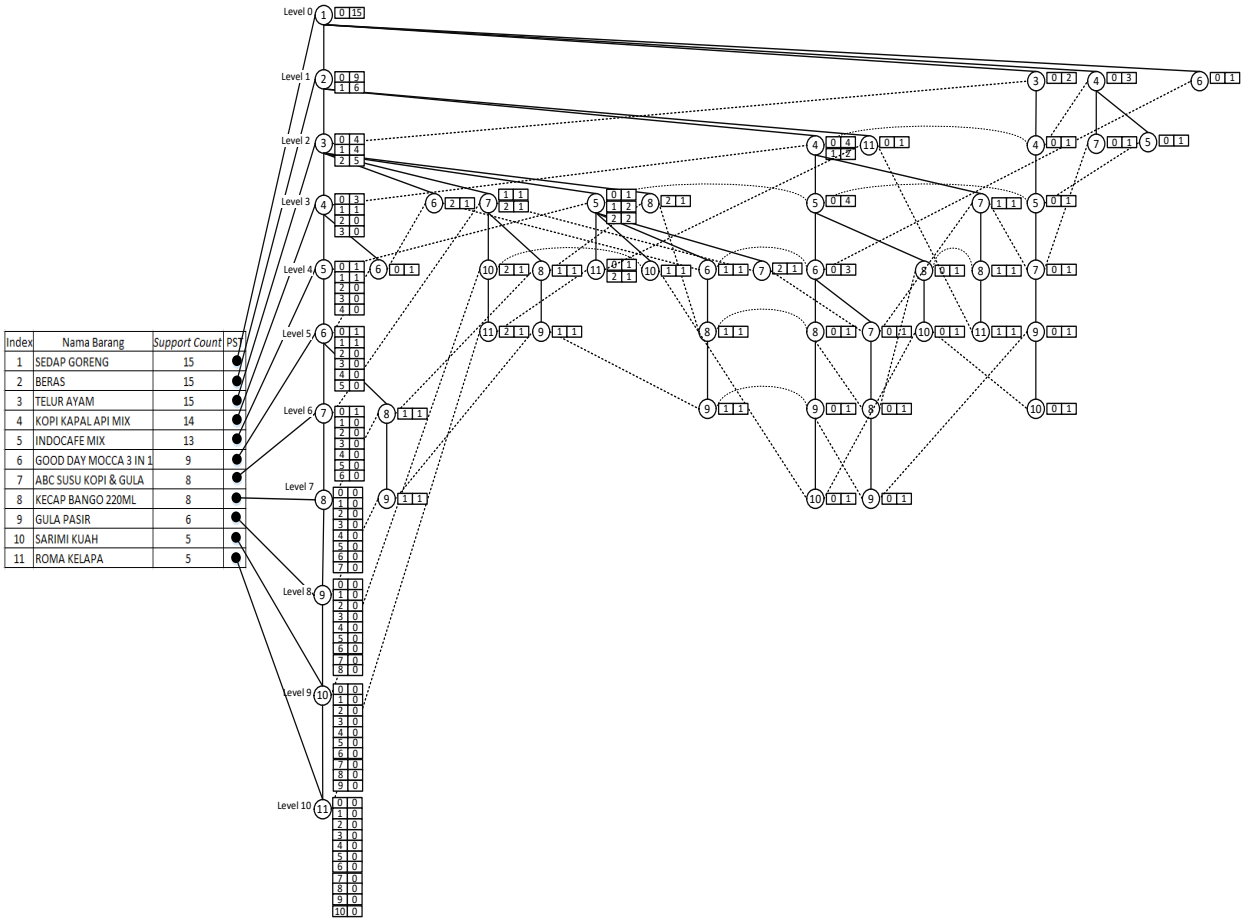
Gambar 3. 4 *CFP-Tree* Transaksi 15080002

3. Nomor Transaksi 15080003 dengan *index* 1 dan 3 pembentukan *CFP-Tree* dengan nomor transaksi 15080002 dapat dilihat pada gambar 3.5 sebagai berikut :



Index	Nama Barang	*Support Count*	PST
1	SEDAP GORENG	15	
2	BERAS	15	
3	TELUR AYAM	15	
4	KOPI KAPAL API MIX	14	
5	INDOCAFE MIX	13	
6	GOOD DAY MOCCA 3 IN 1	9	
7	ABC SUSU KOPI & GULA	8	
8	KECAP BANGO 220ML	8	
9	GULA PASIR	6	
10	SARIMI KUAH	5	
11	ROMA KELAPA	5	

Gambar 3. 5 *CFP-Tree* Transaksi 15080003



Index	Nama Barang	*Support Count*	PST
1	SEDAP GORENG	15	
2	BERAS	15	
3	TELUR AYAM	15	
4	KOPI KAPAL API MIX	14	
5	INDOCAFE MIX	13	
6	GOOD DAY MOCCA 3 IN 1	9	
7	ABC SUSU KOPI & GULA	8	
8	KECAP BANGO 220ML	8	
9	GULA PASIR	6	
10	SARIMI KUAH	5	
11	ROMA KELAPA	5	

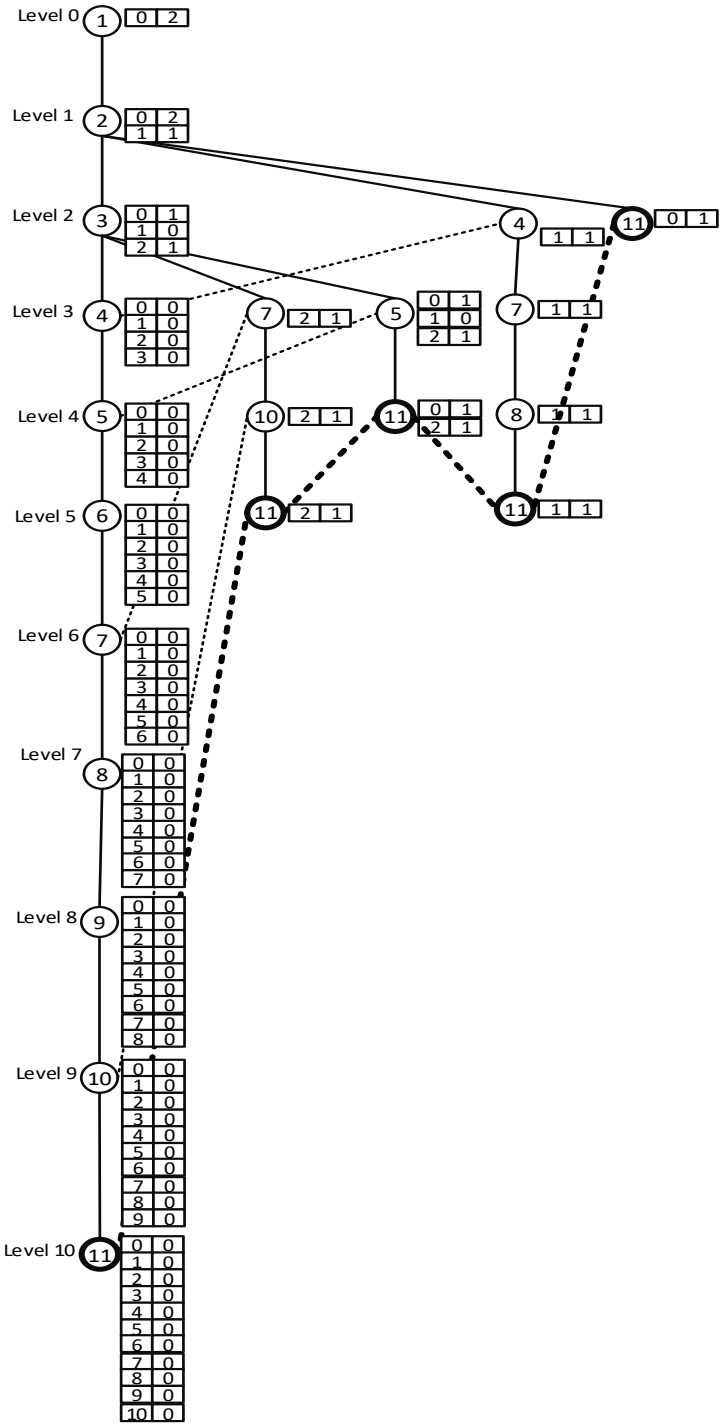
Gambar 3. 6 *Global CFP-Tree*

f. Pengalian frequent patterns

Setelah tahap *CFP-Tree* dari sekumpulan data transaksi, tahap selanjutnya adalah membangun local *CFP-Tree* untuk mencari *frequent item set*. Berikut langkah-langkah algoritma *CT-Pro* :

1. Dimulai dengan *index* yang *support count* terkeci (x) dari *global item table*.
2. Lakukan penelusuran *node* yang berkaitan dengan *index* x dengan menghitung *count* = 1 pada setiap node yang dikunjungi mulai dari *path* ke *root* untuk mencari *node* yang memenuhi *minimum support count*.
3. Masukan setiap *item* yang memenuhi *minimum support* ke *local item table*.
4. Berikan *index* baru pada node yang ada di *local item table* secara *ascending*
5. Setelah local item table terbentuk lakukan pemetaan *index* baru terhadap node yang memenuhi *support* sesuai dengan *index* di *local item table*.
6. Setelah dilakukan pemetaan lakukan pembentukan *local CFP-Tree* dengan aturan yang sama saat pembentukan *global CFP-Tree*.
7. Lakukan pembentukan local *CFP-Tree* projection yaitu dimana x akan menjad root *index* yang ada di local *CFP-Tree* akan menjad anaknya.
8. Lakukan penelusurandai *local CFP-Tree* diulang secara *bottom-up* dari *node* ke-i dengan menghitung *count* dari setiap *node,* jika *node* k memenuhi *minimum support* jadikan k menjadi anak dari *node* ke-i.

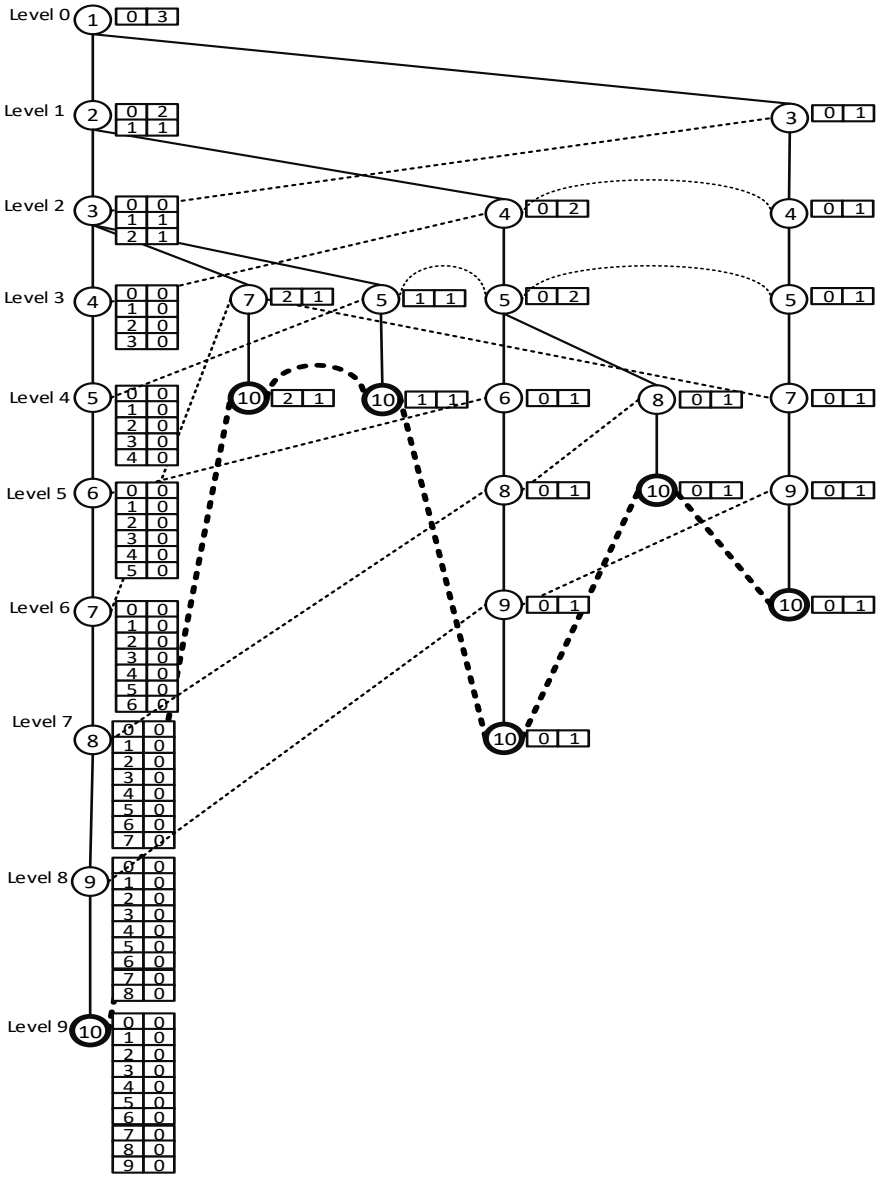
- Untuk pencarian *Local Frequent Item Tree* untuk *index* 11. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya memiliki *index* 11, untuk jelasnya dapat dilihat pada gambar 3.7 di bawah ini:



Gambar 3. 7 *CFP-Tree* Berakhiran *index* 11

Dari gambar 3.7 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(2),2(3),3(2),4(1),5(2),7(2) Karena *minimum support*-nya adalah 5, maka untuk *node* yang tidak memenuhi *minimum support* tidak dimasukan dalam *local item table*. Pada *node* diatas tidak ada yang memenuhi nilai *minimum support*, maka tidak terbentuk *local item table* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

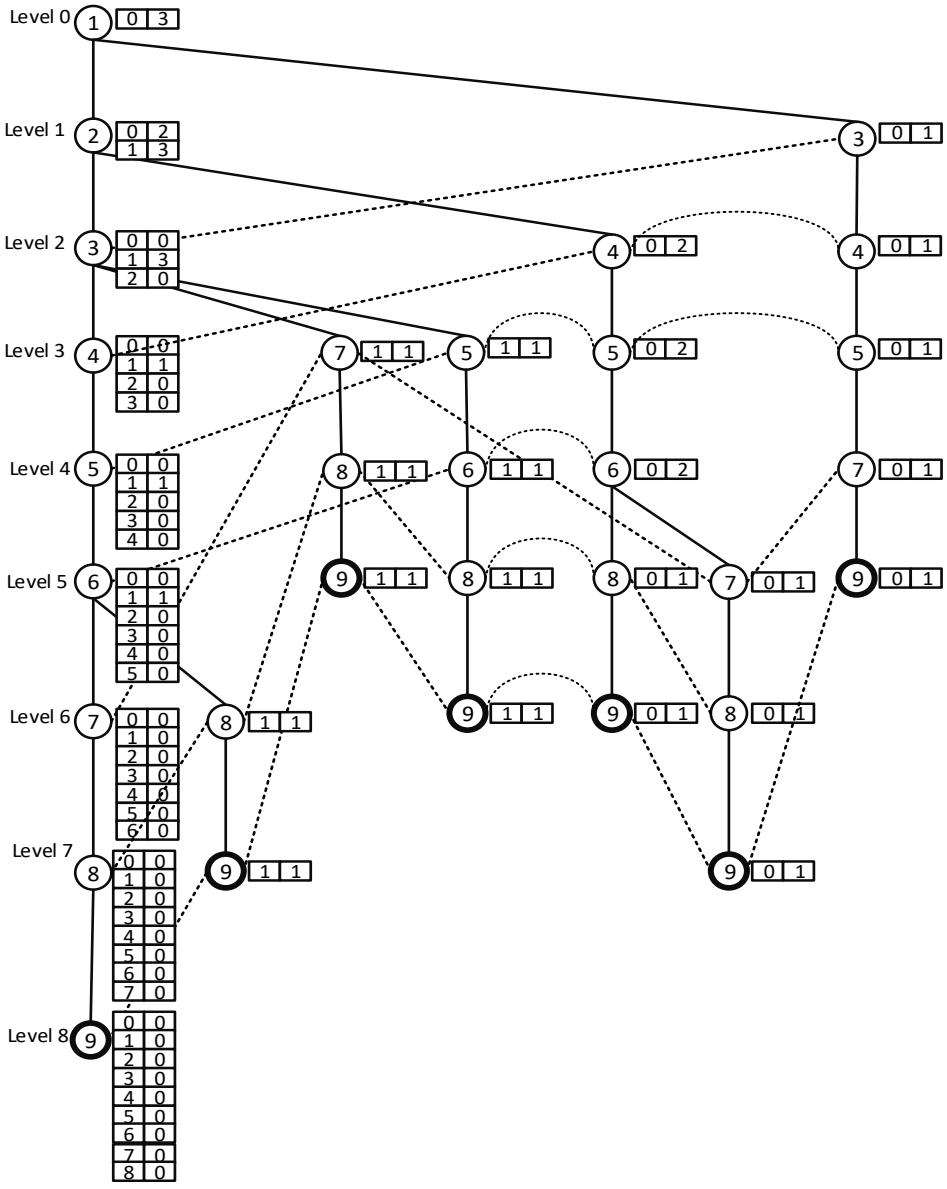
- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 10. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 10, untuk jelasnya dapat dilihat pada gambar 3.8 di bawah ini:



Gambar 3. 8 *CFP-Tree* Berakhiran *index* 10

Dari gambar 3.8 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(3), 2(3), 3(3), 4(2), 5(4), 6(1), 7(2), 8(2), 9(2) Karena *minimum support*-nya adalah 5, maka untuk *node* yang tidak memenuhi *minimum support* tidak dimasukan dalam *local item table*. Pada *node* diatas tidak ada yang memenuhi nilai *minimum support*, maka tidak terbentuk *local item table* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 9. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran index 9, untuk jelasnya dapat dilihat pada gambar 3.9 di bawah ini:



Gambar 3. 9 *CFP-Tree* Berakhiran *index* 9

Berikut penelusuran untuk *index* 9 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.14 dibawah ini:

Tabel 3. 14. Hasil penelusuran *index* 9

No	*Path*
1	Null
2	9,8,6,5,4,3,2
3	9,8,7,3,2
4	9,8,6,5,3,2
5	9,8,6,5,4,2,1
6	9,8,7,6,5,4,2,1
7	9,7,5,4,3,1

Dari gambar 3.14 di atas didapat jumlah *support count* dari setiap *node* yaitu 1(3), 2(5), 3(4), 4(4), 5(5), 6(4), 7(3), 8(5) karena minimum *support*nya adalah 5, maka untuk *node* 1(3), 3(4), 4(4), 6(4), 7(3) tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.41 di bawah ini:

Tabel 3. 15. *Local item table*

Indeks	*Node*	Nama Barang	*Count*
1	2	BERAS	5
2	5	INDOCAFE MIX	5
3	8	KECAP BANGO 220ML	5

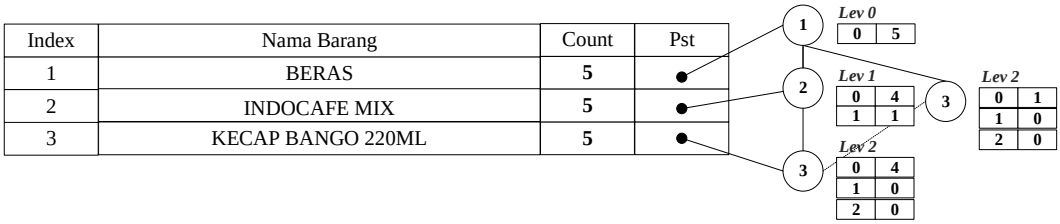
Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.16 di bawah:

Tabel 3. 16. *Table* hasil *mapping*

No	*Path*
1	1,2,3
2	1,3
3	1,2,3
4	1,2,3
5	1,2,3
6	2

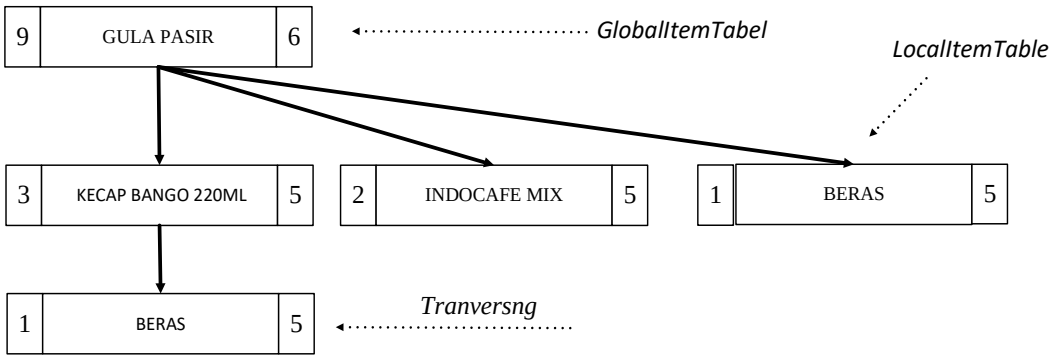
Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan

Global CFP-Tree, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent item*. Berikut adalah *Local CFP-Tree* dari *index* 9 yang dapat dilihat pada gambar 3.10 dibawah ini:



Gambar 3. 10. *Local CFP-Tree Index* 9

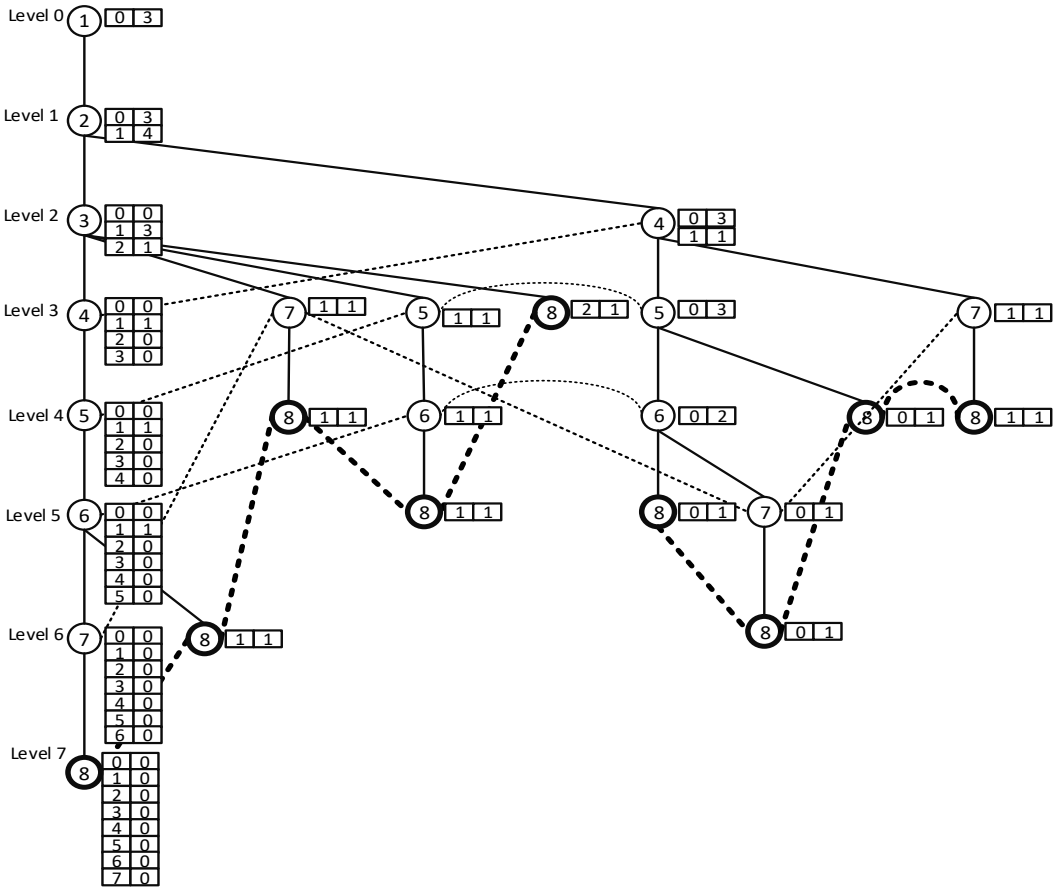
Setelah *Local CFP-Tree index* 9 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 9 akan menjadi *root* dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 9. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran dari *local CFP-Tree index* 9 mulai dari *path* sampai *root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.11 di bawah:



Gambar 3. 11 *Local CFP-Tree Projection index* 9

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 9 adalah sebagai berikut: (9,3:5), (9,3,1;5), (9,2:5), dan (9,1:5) .

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 8 . Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 8, untuk jelasnya dapat dilihat pada gambar 3.12 berikut:



Gambar 3. 12 *CFP-Tree* Berakhiran *index* 8

Berikut penelusuran untuk *index* 8 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.17 dibawah ini:

Tabel 3. 17 hasil penelusuran *index* 8

No	*Path*
1	Null
2	8,6,5,4,3,2
3	8,7,3,2
4	8,6,5,3,2
5	8,3
6	8,6,5,4,2,1
7	8,7,6,5,4,2,1
8	8,5,4,2,1
9	8,7,4,2

Dari gambar 3.12 diatas dapat jumlah *support count* dari setiap *node* yaitu 1(3),2(7),3(4),4(5),5(5),6(4),7(3) karena minimum *support*nya adalah 5, maka untuk node 1(3), 3(4), 6(4), 7(3) tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilhat pada tabel 3.18 di bawah:

Tabel 3. 18. *Local item table*

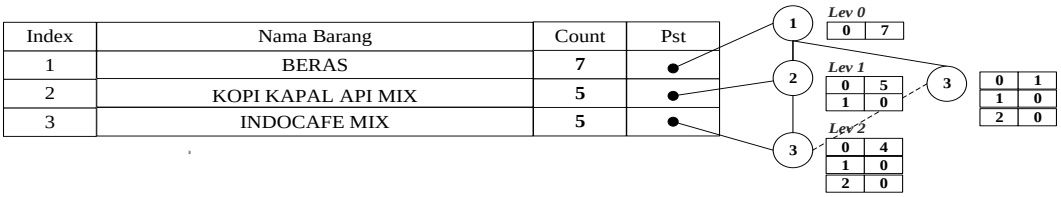
Index	*Node*	Nama Barang	*Count*
1	2	BERAS	7
2	4	KOPI KAPAL API MIX	5
3	5	INDOCAFE MIX	5

Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.19 di bawah:

Tabel 3. 19. *Table* hasil *mapping*

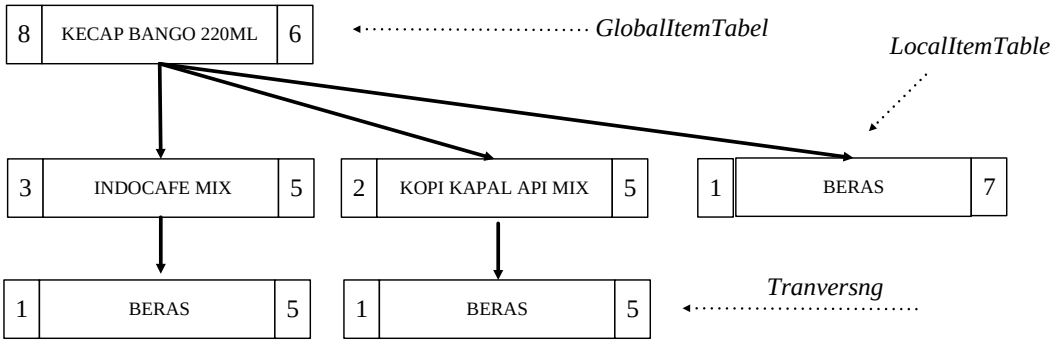
No	*Path*
1	1,2,3
2	1
3	1,3
4	1,2,3
5	1,2,3
6	1,2,3
7	1,2

Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global Ite*m tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent item*. Berikut adalah *Local CFP-Tree* dari *index* 8 dapat dilihat pada gambar 3.14:



Gambar 3. 13 *Local CFP-Tree index* 8

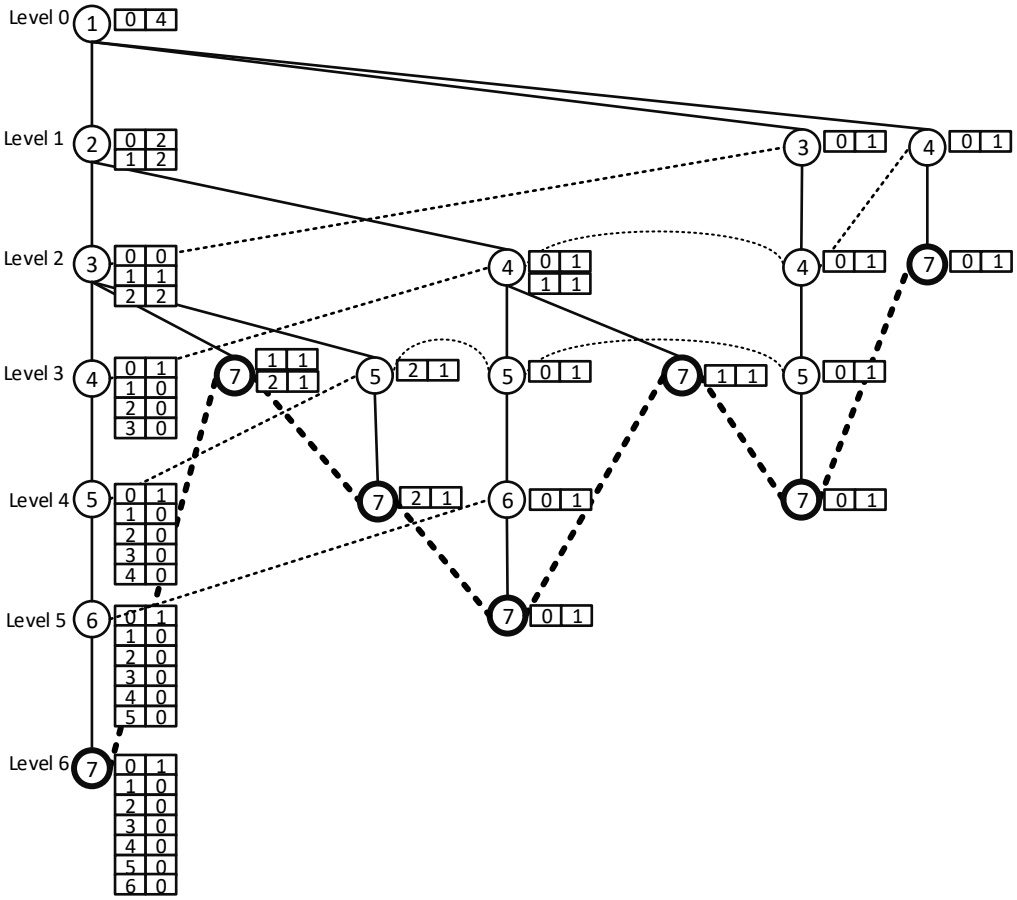
Setelah *Local CFP-Tree index* 8 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 8 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 8. Untuk menemukan local *frequent pattern tree*, lakukan penelusuran dari *local CFP-Tree index* 8 mulai dari *path* sampai *root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.14 berikut:



Gambar 3. 14 *Local CFP-Tree Projection index* 8

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 8 yaitu (8,3;5), (8,3,1;5), (8,2;5), (8,2,1;5), (8,1;7).

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 7. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 7, untuk jelasnya dapat dilihat pada gambar 3.15 berikut:



Gambar 3. 15 *CFP-Tree* Berakhiran *index* 7

Berikut penelusuran untuk *index* 7 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.20 dibawah ini:

Tabel 3. 20 hasil penelusuran *Index* 7

No	*Path*
1	7,6,5,4,3,2,1
2	7,3
3	7,3,2
4	7,5,3
5	7,6,5,4,2,1
6	7,4,2
7	7,5,4,3,1
8	7,4,1

Dari gambar 3.15 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(4), 2(4), 3(4), 4(5),5(4),6 (2) karena *minimum support*nya adalah 5, maka untuk *node* , 1(4), 2(4), 3(4), 5(4), 6(2) tidak memenuhi nilai *minimum support*, maka tidak dimasukan ke dalam *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk pada tabel 3.21 di bawah:

Tabel 3. 21 *Local item table*

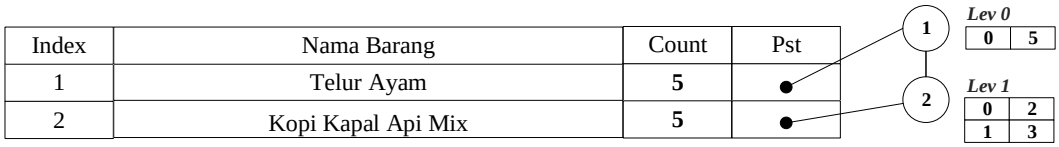
Index	*Node*	Nama Barang	*Count*
1	3	Telur Ayam	5
2	4	KOPI KAPAL API MIX	5

Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.22 di bawah:

Tabel 3. 22 *table* hasil *mapping*

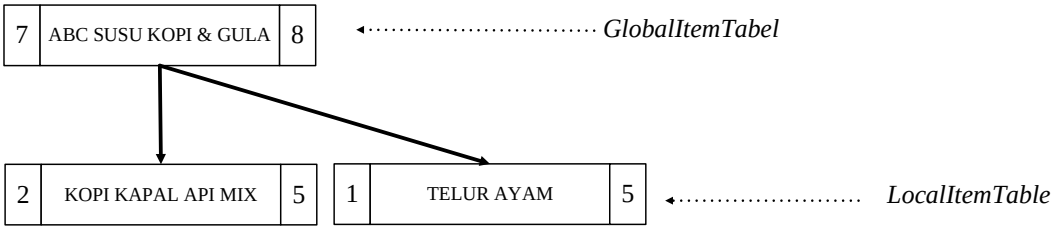
No	*Path*
1	1,2
2	1
3	1
4	1
5	2
6	2
7	1,2
8	2

Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent* item. Berikut adalah *Local CFP-Tree* dari *index* 7 yang dapat dilihat pada gambar 3.16 :



Gambar 3. 16 *Local CFP-Tree index* 7

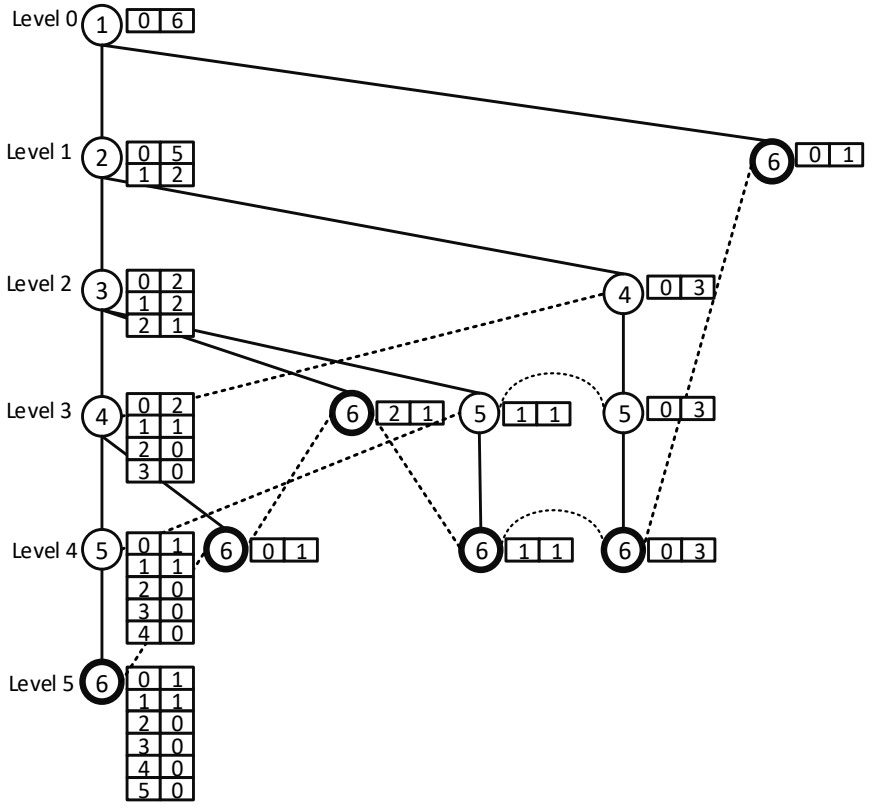
Setelah *Local CFP-Tree index* 7 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 7 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 7. Untuk menemukan local *frequent pattern tree,* lakukan penelusuran dari *local CFP-Tree index* 7 mulai dari *path* sampai *root* secara *bottom-up,* untuk lebih jelasnya dapat dilihat pada gambar 3.17 berikut:



Gambar 3. 17 *Local CFP-Tree Projection index* 7

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 7 dengan nama barang Abc susu kopi dan gula yaitu (7,2;5) dan (7,1;5)

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 6. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya *index* 6, untuk jelasnya dapat dilihat pada gambar 3.18 berikut:



Gambar 3. 18 *CFP-Tree* Berakhiran *index* 6

Berikut penelusuran untuk *index* 6 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.19 dibawah ini:

Tabel 3. 23 hasil penelusuran *index* 6

No	*Path*
1	6,5,4,3,2
2	6,5,4,3,2,1
3	6,4,3,2,1
4	6,3
5	6,5,3,2
6	6,5,4,2,1
7	6,5,4,2,1
8	6,5,4,2,1
9	6,1

Dari gambar 3.19 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(6), 2(7), 3(5), 4(6),5(7) karena *minimum support*nya adalah 5, maka semua *node* masuk ke dalam *Local Item Table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.24 di bawah:

Tabel 3. 24 *local item table*

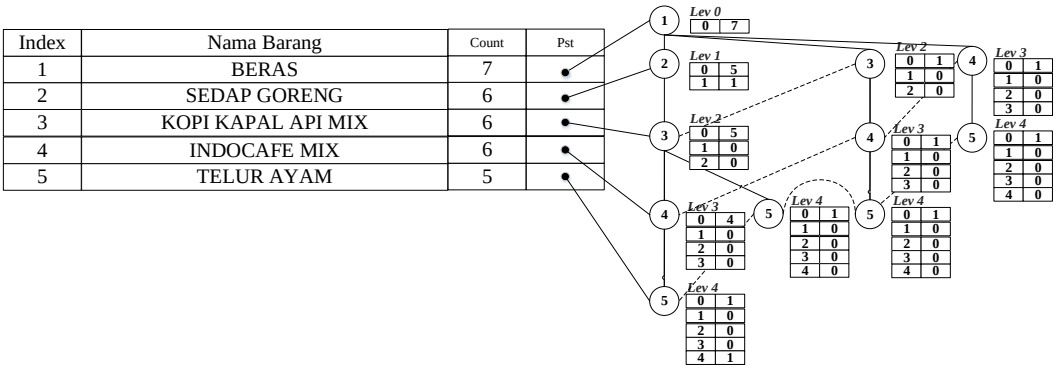
Index	*Node*	Nama Barang	*Count*
1	2	BERAS	7
2	1	SEDAP GORENG	6
3	4	KOPI KAPAL API MIX	6
4	5	INDOCAFE MIX	6
5	3	TELUR AYAM	5

Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.25 di bawah:

Tabel 3. 25 *table* hasil *mapping*

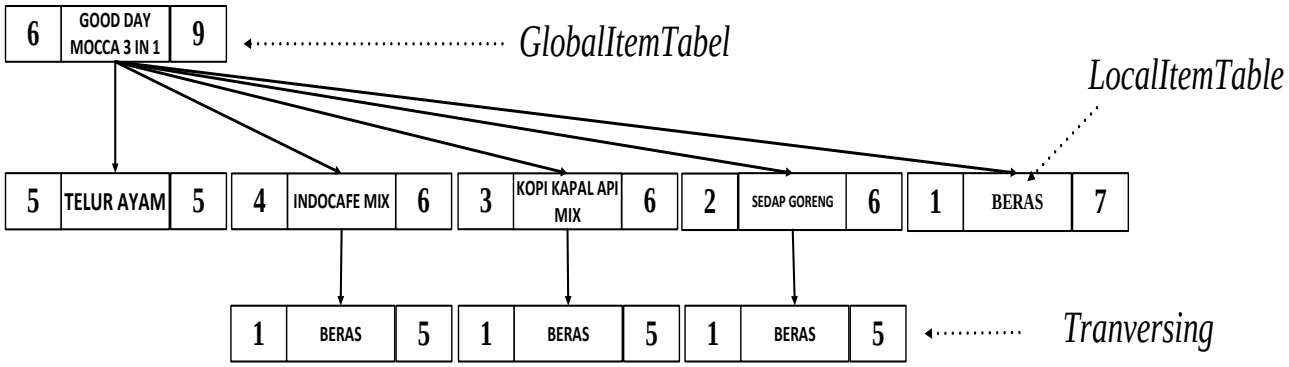
No	*Path*
1	1,3,4,5
2	1,2,3,4,5
3	1,2,3,5
4	5
5	1,4,5
6	1,2,3,4
7	1,2,3,4
8	1,2,3,4
9	2

Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent* item. Berikut adalah *Local CFP-Tree* dari *index* 6 dapat dilihat pada gambar 3.19:



Gambar 3. 19 *Local CFP-Tree index* 6

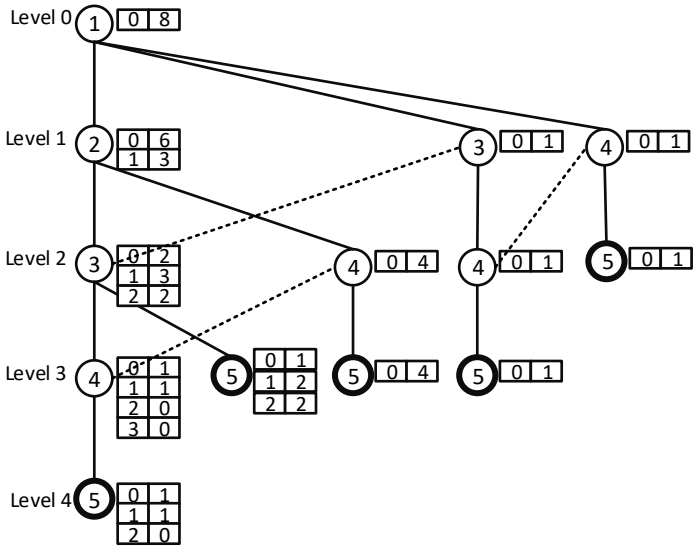
Setelah *Local CFP-Tree index* 6 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 6 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 6. Untuk menemukan local *frequent pattern tree,* lakukan penelusuran dari *local CFP-Tree index* 6 mulai dari *path* sampai *root* secara *bottom-up,* untuk lebih jelasnya dapat dilihat pada gambar 3.21 berikut:



Gambar 3. 20 *Local CFP-Tree Projection index* 6

Dari *Local CFP-Tree projection* maka didapatkan item frequent untuk *index* 6 dengan nama barang good day mocca 3 in 1 adalah sebagai berikut: (6,5;5), (6,4;6), (6,4,1;5), (6,3;6), (6,3,1;5), (6,2;6), (6,2,1;6), (6,1;7).

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 5. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 5, untuk jelasnya dapat dilihat pada gambar 3.22 berikut:



Gambar 3. 21 *CFP-Tree* Berakhiran *index* 5

Berikut penelusuran untuk *index* 5 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.26 dibawah ini:

Tabel 3. 26 hasil penelusuran *index* 5

No	*Path*
1	5,4,3,2
2	5,4,3,2,1
3	5,3
4	5,3
5	5,3,2
6	5,3,2.
7	5,3,2,1
8	5,4,2,1
9	5,4,2,1
10	5,4,2,1
11	5,4,2,1
12	5,4,3,1
13	5,4,1

Dari gambar 3.22 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(8), 2(9), 3(8), 4(8) karena *minimum support*nya adalah 5, maka semua *node* masuk ke dalam *Local Item Table* yaitu penomoran secara ascending dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.27 :

Tabel 3. 27 *local item table*

Index	*Node*	Nama Barang	*Count*
1	2	BERAS	9
2	3	TELUR AYAM	8
3	4	KOPI KAPAL API MIX	8
4	1	SEDAP GORENG	8

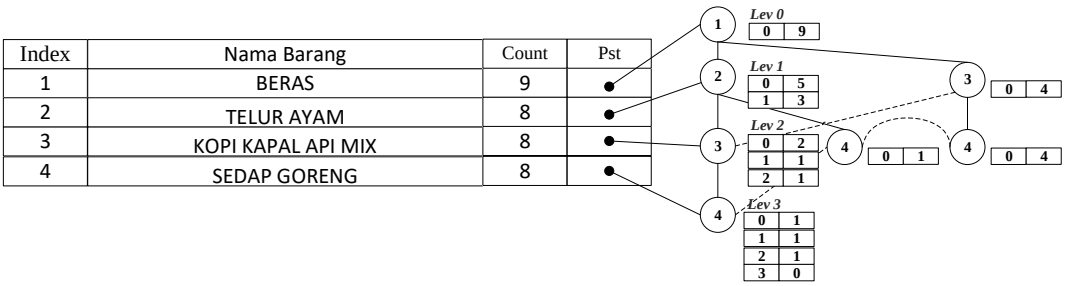
Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.28 di bawah:

Tabel 3. 28 *table* hasil mapping

No	*Path*
1	1,2,3
2	1,2,3,4
3	2
4	2
5	1,2
6	1,2
7	1,2,4
8	1,3,4
9	1,3,4
10	1,3,4
11	1,3,4
12	2,3,4
13	3,4

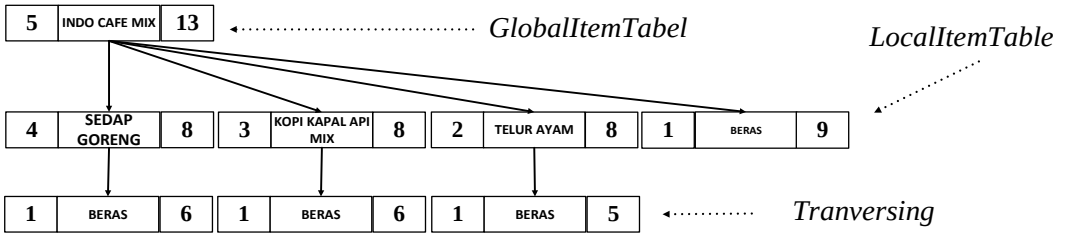
Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item tabel* yang terbentuk

dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent* item. Berikut adalah *Local CFP-Tree* dari *index* 5 dapat dilihat pada gambar 3.23:



Gambar 3. 22 *Local CFP-Tree index* 5

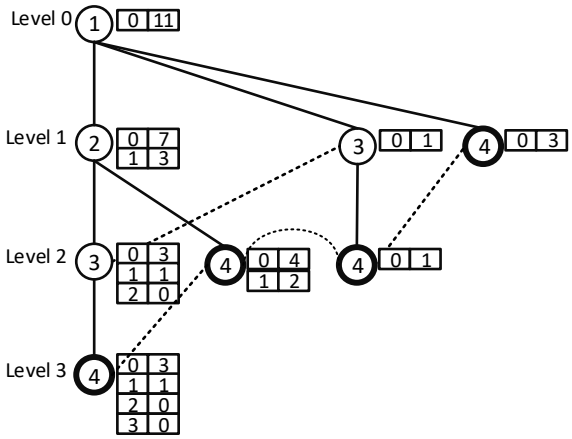
Setelah *Local CFP-Tree index* 5 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 5 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 5. Untuk menemukan local *frequent pattern tree*, lakukan penelusuran dari *local CFP-Tree index* 5 mulai dari *path* sampai *root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.24 berikut:



Gambar 3. 23 *Local CFP-Tree Projection index* 5

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 5 dengan nama barang indo café mix yaitu (5,4;8), (5,4,1;6) (5,3;8), (5,3,1;6), (5,2;8), (5,2,1;5), (5,1;9)

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 4. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 4, untuk jelasnya dapat dilihat pada gambar 3.24 berikut:



Gambar 3. 24 *CFP-Tree* Berakhiran *index* 4

Berikut penelusuran untuk *index* 4 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.29 dibawah ini:

Tabel 3. 29 penelusuran *index* 4

No	*Path*
1	4,3,2
2	4,3,2,1
3	4,3,2,1
4	4,3,2,1
5	4,2
6	4,2
7	4,2,1
8	4,2,1
9	4,2,1
10	4,2,1
11	4,3,1
12	4,1
13	4,1
14	4,1

Dari gambar 3.24 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(11), 2(10), 3(8) karena *minimum support*nya adalah 5, maka semua *node* masuk ke dalam *Local Item Table* yaitu penomoran secara *ascending* dari jumlah

support count terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.30 di bawah ini:

Tabel 3. 30 *local item table*

Index	*Node*	Nama Barang	*Count*
1	1	SEDAP GORENG	11
2	2	BERAS	10
3	3	TELUR AYAM	5

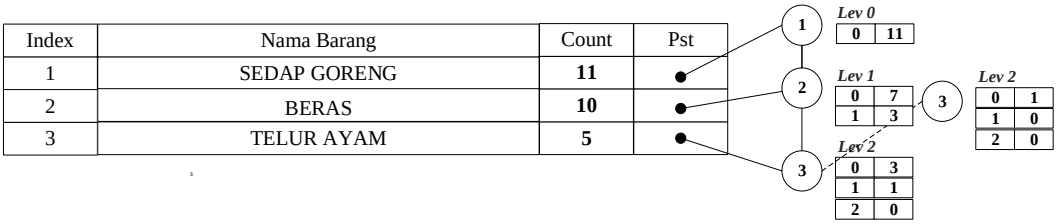
Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.31 di bawah:

Tabel 3. 31 tabel hasil *mapping*

No	*Path*
1	2,3
2	1,2,3
3	1,2,3
4	1,2,3
5	2
6	2
7	1,2
8	1,2
9	1,2
10	1,2
11	1,3
12	1
13	1
14	1

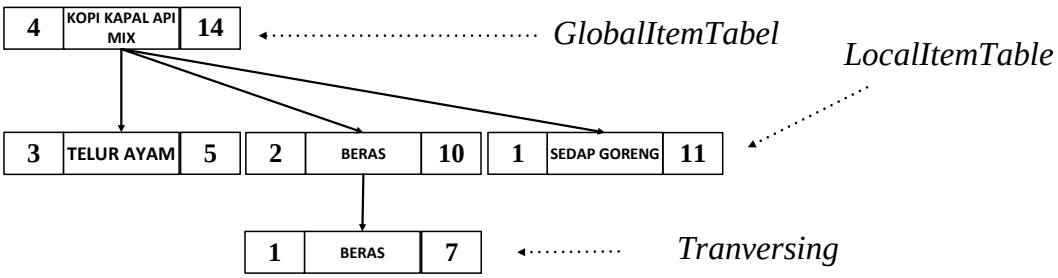
Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global Item* tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local*

Frequent item. Berikut adalah *Local CFP-Tree* dari *index* 4 dapat dilihat pada gambar 3.25:



Gambar 3. 25 *Local CFP-Tree index* 4

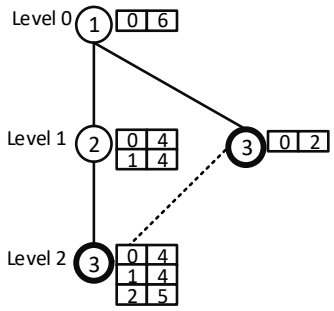
Setelah *Local CFP-Tree index* 4 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 4 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 4. Untuk menemukan local *frequent pattern tree*, lakukan penelusuran dari *local CFP-Tree index* 4 mulai dari *path* sampai *root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.26 di bawah:



Gambar 3. 26 *Local CFP-Tree Projection index* 4

Dari *Local CFP-Tree projection* maka didapatkan item frequent *index* 4 dengan nama barang kopi kapal api mix adalah sebagai berikut: (4,3;5), (4,2,;10), (4,2,1;7), (4,1;11)

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 3. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 3, untuk jelasnya dapat dilihat pada gambar 3.27 berikut:



Gambar 3. 27 CFP-Tree Berakhiran *index* 3

Berikut penelusuran untuk *index* 3 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.32 dibawah ini:

Tabel 3. 32 Hasil penelusuran *index* 3

No	*Path*
1	3,2
2	3,2
3	3,2
4	3,2
5	3,2,1
6	3,2,1
7	3,2,1
8	3,2,1
9	3,1
10	3,1

Dari gambar 3.27 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(6), 2(8) karena *minimum support*nya adalah 5. maka semua *node* masuk ke dalam *Local Item Table* yaitu penomoran secara ascending dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.33 di bawah:

Tabel 3. 33 *local item table*

Index	*Node*	Nama Barang	*Count*
1	2	BERAS	8
2	1	SEDAP GORENG	6

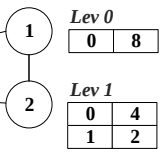
Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.34:

Tabel 3. 34 *table* hasil *mapping*

No	*Path*
1	1
2	1
3	1
4	1
5	1,2
6	1,2
7	1,2
8	1,2
9	2
10	2

Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item* tabel yang terbentuk dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent* item. Berikut adalah *Local CFP-Tree* dari *index* 3 dapat dilihat pada gambar 3.28 di bawah:

Index	Nama Barang	Count	Pst
1	BERAS	**8**	
2	SEDAP GORENG	**6**	



Gambar 3. 28 *Local CFP-Tree index* 3

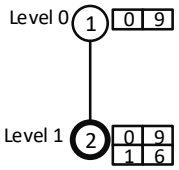
Setelah *Local CFP-Tree index* 3 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 3 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 3. Untuk menemukan local *frequent pattern tree*, lakukan penelusuran dari *local CFP-Tree index* 3 mulai dari *path* sampai *root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.29 berikut:



Gambar 3. 29 *Local CFP-Tree projection index* 3

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 3 dengan nama barang telur ayam adalah sebagai berikut: (3,2;6), (3,1;8)

- Untuk pencarian *Local Frequent Pattern Tree* untuk *index* 2. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 2, untuk jelasnya dapat dilihat pada gambar 3.30 di bawah:



Gambar 3. 30 *CFP-Tree* Berakhiran *index* 2

Berikut penelusuran untuk *index* 2 pada *global CFP-Tree* yang dapat dilihat pada tabel 3.35 dibawah ini:

Tabel 3. 35 hasil penelusuran *index* 2

No	*Path*
1	2,1
2	2,1
3	2,1
4	2,1
5	2,1
6	2,1
7	2,1
8	2,1
9	2,1

Dari gambar 3.30 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(9) karena *minimum support*nya adalah 5, maka semua *node* masuk ke dalam *Local Item Table* maka semua *node* masuk ke dalam *Local Item Table* yaitu

penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk dapat dilihat pada tabel 3.36:

Tabel 3. 36 *local item table*

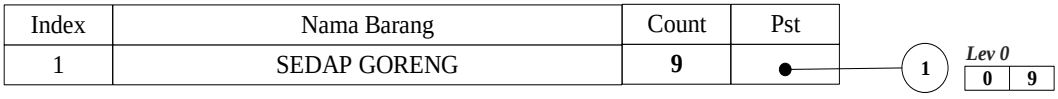
index	*Node*	Nama Barang	*Count*
1	1	SEDEAP GORENG	9

Setelah *local item table* terbentuk, maka tahapan selanjutnya adalah memetakan node yang memenuhi *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.33 di bawah:

Tabel 3. 37 *table* hasil *mapping*

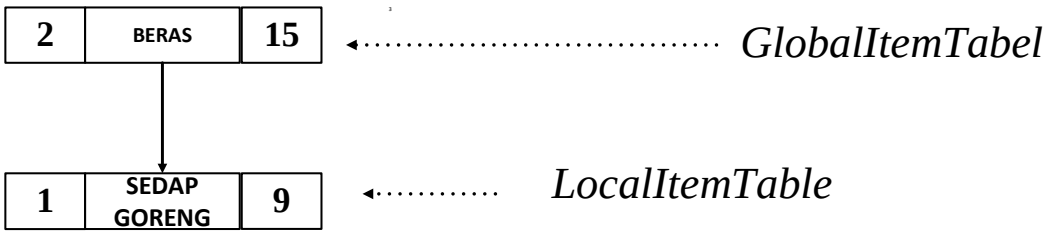
No	*Path*
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

Setelah proses *mapping*, selanjutnya melakukan pembuatan *Local CFPTree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global* Item tabel yang terbentuk dari *Global* Item tabel data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item* tabel yang terbentuk dari *Local Frequent* item. Berikut adalah *Local CFP-Tree* dari *index* 2 dapat dilihat pada gambar 3.31 di bawah:



Gambar 3. 31 *Local CFP-Tree index* 2

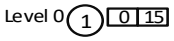
Setelah *Local CFP-Tree index* 2 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* dimana *index* 2 akan menjadi root dan setiap *index* di *local CFP-Tree* akan menjadi anak dari 2. Untuk menemukan local *frequent pattern tree,* lakukan penelusuran dari *local CFP-Tree index* 2 mulai dari *path* sampai *root* secara *bottom-up,* untuk lebih jelasnya dapat dilihat pada gambar 3.32 di bawah:



Gambar 3. 32 *Local CFP-Tree projection index* 2

Dari *Local CFP-Tree projection* maka didapatkan item frequent untuk *index* 2 dengan nama barang beras adalah sebagai berikut: (2,1;9).

- Untuk pencarian *Local Frequent Pattern Tree index* 1. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran *index* 1 , untuk jelasnya dapat dilihat pada gambar 3.33 di bawah:



Gambar 3. 33 *CFP-Tree* Berakhiran *index* 1

Dari gambar 3.33 diatas tidak ada *node* yang ditemukan, maka tidak terbentuk *Local Item Table* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

2. **Mendefinisikan *Condition* dan *Result* (*Generate Rule*)**

Dari *frequent pattern* yang didapat, akan di-*generate* untuk mendapatkan *rule* dengan *minimum confidence* = 65%. Dihitung dengan persamaan 2.4 sebagai berikut :

$$\text{Confidence} = (A \rightarrow B) = \frac{\text{Jumlah Transaksi Yang Mengandung A\&B}}{\text{Jumlah Transaksi Yang Mengandung A}} \text{x}100\%$$

(Persamaan 2-4)

Berikut adalah hasil *generate rule* diacu pada table 3.38 dibawah ini :

Tabel 3. 38 Hasil *Generate Rule*

Frequent Itemset	*Confidence*
9 → 3	(5/6) * 100% = 83,33%
9 → 3,1	(5/6) * 100% = 83,33%
9 → 2	(5/6) * 100% = 83,33%
9 → 1	(5/6) * 100% = 83,33%
8 → 3	(5/8) * 100% = 62,5%
8 → 3,1	(5/8) * 100% = 62,5%
8 → 2	(5/8) * 100% = 62,5%
8 → 2,1	(5/8) * 100% = 62,5%
8 → 1	(7/8) * 100% = 87,5%
7 → 2	(5/8) * 100% = 62,5%
7 → 1	(5/8) * 100% = 62,5%
6 → 5	(5/9) * 100% = 55,56%
6 → 4	(6/9) * 100% = 66.67%
6 → 4, 1	(6/9) * 100% = 66,67%
6 → 3	(6/9) * 100% = 66,67%
6 → 3,1	(6/9) * 100% = 66,67%
6 → 2	(6/9) * 100% = 66.67%
6 → 2,1	(5/9) * 100% = 55.56%
6 → 1	(7/9) * 100% = 77,78%
5 → 4	(8/13) * 100% = 61.53%
5 → 4, 1	(8/13) * 100% = 61.53%
5 → 3	(8/13) * 100% = 61.53%
5 → 3, 1	(6/13) * 100% = 46.15%
5 → 2	(8/13) * 100% = 61.53%
5 → 2,1	(5/13) * 100% = 38.46%
5 → 1	(9/13) * 100% = 69.23%
4 → 3	(5/14) * 100% = 35.71%
4 → 2	(10/14) * 100% = 71.42%

Frequent Itemset	*Confidence*
4 → 2,1	(7/14) * 100% = 50%
4 → 1	(11/14) * 100% = 78,57%
3 → 2	(6/15) * 100% = 40%
3 → 1	(8/15) * 100% = 53,33%
2 → 1	(9/15) * 100% = 60%

Setelah perhitungan *confidence* diketahui maka selanjutnya dilakukan pengecekan, jika tidak memenuhi *minimum confidence* barang tersebut akan dihapus untuk menemukan rule. Setiap *id global item* akan dikembalikan sesuai dengan nama barang agar pembacaan rule lebih jelas. Berikut hasil penentuan paket barang dalam bentuk awal dapat dilihat pada tabel 3.39 di bawah :

Tabel 3. 39 Hasil *Rule* yang terbentuk

Frequent Itemset	Nama Barang	*Confidence*
9 → 3	Gula Pasir → Kecap Bango 220ML	(5/6) * 100% = 83,33%
9 → 3,1	Gula Pasir → Kecap Bango 220ML,Beras	(5/6) * 100% = 83,33%
9 → 2	Gula Pasir → Indocafe mix	(5/6) * 100% = 83,33%
9 → 1	Gula Pasir → Beras	(5/6) * 100% = 83,33%
8 → 3	Kecap Bango 220ML → Indocafe Mix	(5/8) * 100% = 62,5%
8 → 3,1	Kecap Bango 220ML → Indocafe Mix, Beras	(5/8) * 100% = 62,5%
8 → 2	Kecap Bango 220ML → Kopi Kapal Api Mix	(5/8) * 100% = 62,5%
8 → 2,1	Kecap Bango 220ML → Kopi Kapal Api Mix Beras	(5/8) * 100% = 62,5%
8 → 1	Kecap Bango 220ML → Beras	(7/8) * 100% = 87,5%
7 → 2	Abc Susu Kopi & Gula → Kopi Kapal Api Mix	(5/8) * 100% = 62,5%
7 → 1	Abc Susu Kopi & Gula → Telur Ayam	(5/8) * 100% = 62,5%
6 → 4	Good Day Mocca 3 In 1 → Indocafe Mix	(6/9) * 100% = 66.67%
6 → 4,1	Good Day Mocca 3 In 1 → Indocafe Mix, Beras	(6/9) * 100% = 66.67%
6 → 3	Good Day Mocca 3 In 1 → Kopi Kapal Api Mix	(6/9) * 100% = 66,67%
6 → 3,1	Good Day Mocca 3 In 1 → Kopi Kapal Api Mix, Beras	(6/9) * 100% = 66,67%
6 → 2	Good Day Mocca 3 In 1 → Beras	(7/9) * 100% = 77,78%
6 → 1	Good Day Mocca 3 In 1 → Sedap Goreng	(6/9) * 100% = 66,67%
5 → 4	Indocafe Mix → Sedap Goreng	(8/13) * 100% = 61.53%
5 → 3	Indocafe Mix → Kopi Kapal Api Mix	(8/13) * 100% = 61.53%
5 → 2	Indocafe Mix → Telur Ayam	(8/13) * 100% = 61.53%
5 → 1	Indocafe Mix → Beras	(9/13) * 100% = 69.23%

Frequent Itemset	Nama Barang	*Confidence*
4 → 2	Kopi Kapal Api Mix → Beras	(10/14) * 100% = 71.42%
4 → 1	Kopi Kapal Api Mix → Sedap Goreng	(11/14) * 100% = 78,57%
2 → 1	Beras → Sedap Goreng	(9/15) * 100% = 60%

Tabel 3. 40 Paket Barang Dalam Bentuk awal

Frequent Itemset	Nama Barang	Harga
9 → 3	Gula Pasir → Kecap Bango 220ML	Rp.20.000
9 → 3,1	Gula Pasir → Kecap Bango 220ML,Beras	Rp.36.100
9 → 2	Gula Pasir → Indocafe mix	Rp.16.300
9 → 1	Gula Pasir → Beras	Rp.27.600
8 → 3	Kecap Bango 220ML → Indocafe Mix	Rp.13.300
8 → 3,1	Kecap Bango 220ML → Indocafe Mix, Beras	Rp.29.400
8 → 2	Kecap Bango 220ML → Kopi Kapal Api Mix	Rp.21.150
8 → 2,1	Kecap Bango 220ML → Kopi Kapal Api Mix Beras	Rp.37.250
8 → 1	Kecap Bango 220ML → Beras	Rp,24.600
7 → 2	Abc Susu Kopi & Gula → Kopi Kapal Api Mix	Rp.21.950
7 → 1	Abc Susu Kopi & Gula → Telur Ayam	Rp.25.300
6 → 4	Good Day Mocca 3 In 1 → Indocafe Mix	Rp.14.400
6 → 4,1	Good Day Mocca 3 In 1 → Indocafe Mix, Beras	Rp.30.500
6 → 3	Good Day Mocca 3 In 1 → Kopi Kapal Api Mix	Rp.22.250
6 → 3,1	Good Day Mocca 3 In 1 → Kopi Kapal Api Mix, Beras	Rp.38.350
6 → 2	Good Day Mocca 3 In 1 → Beras	Rp.25.700
6 → 1	Good Day Mocca 3 In 1 → Sedap Goreng	Rp.11.100
5 → 4	Indocafe Mix → Sedap Goreng	Rp.9.300
5 → 3	Indocafe Mix → Kopi Kapal Api Mix	Rp.17.450
5 → 2	Indocafe Mix → Telur Ayam	Rp.20.800
5 → 1	Indocafe Mix → Beras	Rp.20.900
4 → 2	Kopi Kapal Api Mix → Beras	Rp.28.750
4 → 1	Kopi Kapal Api Mix → Sedap Goreng	Rp.14.150
2 → 1	Beras → Sedap Goreng	Rp.17.500

Dalam penentuan kombinasi yang akan ditawarkan dalam bentuk paket, Mini Market Barokah memiliki pertimbangan, yaitu:

1. Jumlah barang yang dipaketkan yaitu sebanyak 3 sampai dengan 5 barang.
2. Harga Paket Minimal sebesar 20.000 dan maksimal 40.000

3. Jumlah paket barang yang terbentuk dalam 1 periode yaitu sebanyak 3 sampai 5 paket
4. Pergantian paket barang dilakukan setiap periode (perbulan)

Berikut merupakan paket barang yang terbentuk dapat dilihat pada tabel 3.41:

Tabel 3. 41 Paket Barang

Paket	Nama Barang	Harga
1	1. Gula Pasir (1 kg) 2.Kecap Bango 220ML (1 pcs) 3.Beras (1 kg)	Rp.36.100
2	1.Kecap Bango 220ML (1 pcs) 2.Indocafe Mix (1 rcg) 3.Beras (1 kg)	Rp.29.400
3	1.Kecap Bango 220ML (1 pcs) 2.Kopi Kapal Api Mix (1 rcg) 3.Beras (1 kg)	Rp.37.250
4	1.Good Day Mocca 3 In 1 (1rcg) 2.Indocafe Mix (1 rcg) 3.Beras (1 kg)	Rp.30.500
5	1.Good Day Mocca 3 In 1 (1 rcg) 2.Kopi Kapal Api Mix (1 rcg) 3.Beras (1 kg)	Rp.38.350

3.1.3 Analisis Non Fungsional dan Kebutuhan Non Fungsional

Analisis *non fungsional* dan kebutuhan *non fungsional* dibagi menjadi beberapa tahap analisis, yaitu analisis pengguna, analisis perangkat lunak dan analisis perangkat keras.

3.1.3.1 Analisis Pengguna

User atau pengguna yang akan mengoperasikan aplikasi ini merupakan manager yang bertindak sebagai pengolah bisnis. Analisis Pengguna dapat dilihat pada table 3.42 di bawah:

Tabel 3. 42 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.

3.1.3.2 Analisis Perangkat Lunak

Analisis kebutuhan perangkat lunak dalam pembangunan aplikasi digunakan untuk mengetahui dan mengoptimalkan implementasi dari aplikasi yang akan dibangun. Berikut merupakan analisis terhadap perangkat lunak dibagian manager mini market barokah:

1. Analisis spesifikasi perangkat lunak pada sistem yang sedang berjalan di bagian manager mini market barokah adalah:
   a. Sistem Operasi *Windows* 7
   b. *Microsoft Office* 2010
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 bagian manager mini market barokah, spesifikasi perangkat lunak yang digunakan masih kurang. Untuk mengatasi hal tersebut akan diinstall MySQL dan juga *Visual Studio* pada komputer yang digunakan.

3.1.3.3 Analisis Perangkat Keras

Analisis kebutuhan perangkat keras digunakan untuk memudahkan proses perancangan dan implementasi. Berikut merupakan analisis terhadap perangkat keras di bagian manager mini market barokah.

1. Analisis spesifikasi perangkat keras pada *system* yang sedang berjalan dibagian manager mini market barokah adalah:
   a. *Processor*: Intel Core i3-550 3.2 Ghz
   b. *Hard disk*: 500 GB
   c. *Memory*: 4 GB
2. Analisis spesifikasi kebutuhan perangkat keras untuk menjalankan sistem yang akan dibangun adalah
   a. *Processor*: berkecepatan minimal 2 GHz
   b. *Hard disk* 500 MB untuk menyimpan *file*
   c. *Memory*: 1 GB
3. Evaluasi kebutuhan perangkat lunak

   Berdasarkan analisis spesifikasi perangkat keras yang ada di bagian *manager* mini market barokah, perangkat keras yang digunakan sudah cukup untuk dapat menjalankan aplikasi *Data mining*.

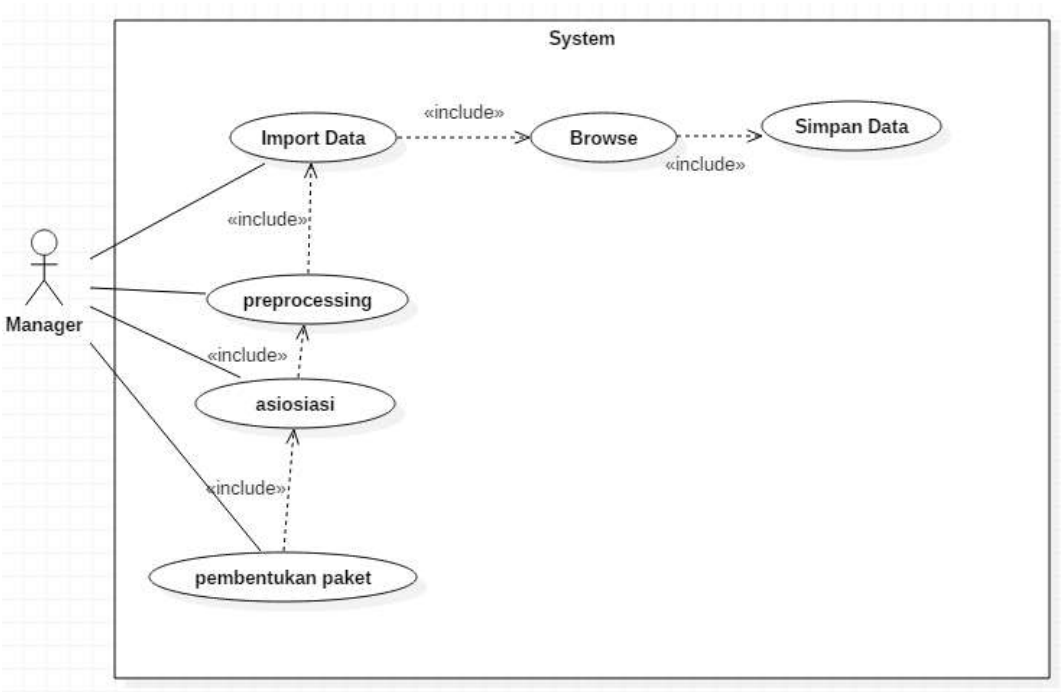
3.1.4. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah analisis terhadap kebutuhan secara fungsional baik dalam aliran data ataupun informasi. Analisis kebutuhan data fungsional yang digunakan dalam sistem ini di gambarkan dengan analisis

pemograman berorientasi objek dengan menggunakan konsep UML (*Unifield Modeling languages*).

3.1.4.1 Diagram *Use Case*

Use case atau diagram *Use case* merupakan pemodelan kelakuan (behavior) sistem yang akan dibuat. Diagram *Use case* dapat dilihat pada gambar 3.34 dan penjelasan *user* dan *use case* dilihat pada tabel 3.43 dan 3.44



Gambar 3. 34 Diagram *Use Case*

Tabel 3. 43 Tabel definisi *user*

User	**Deskripsi**
Manager	*Manager* bertugas untuk melakukan import data, memasukan *minimum support* dan *minimum confidance*

Tabel 3. 44 Tabel 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 disimpan kedalam *database*.
2	*Browse*	*User* memilih data transaksi
3	*Import*	*User* mengimportkan data transaksi ke dalam *database*
4	*Processing*	Data transaksi yang sudah tersimpan di dalam *database* kemudian akan dilakukan proses pembersihan data, pemilihan atribut dan hitung *support count*
5	Asosiasi	*User* menginputkan nilai *Minimum Support* dan *Minimum Confidence*, kemudian sistem melakukan pencarian *frequent itemset,* melakukan proses eliminasi, menghasilkan *rule* dan informasi rekomendasi paket.
6	Pembentukan Paket	*Manager* memasukan nilai parameter yaitu nilai minimal dan maksimal harga yang memenuhi batasan,melakukan filter, menghasilkan informasi paket

3.1.4.2 Skenario Use Case

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* pada tabel 3.45 di bawah:

Tabel 3. 45 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. 46 Skenario *Use Case Import* Data

Use case Name	*Import Data*	
Related Requirements	Requirement *A.1*	
Goal In Context	*Import* data transaksi Mini Market Barokah 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	*Manager* melakukan *import data*
	2	Sistem menampilkan *import* data
	3	*Manager* klik browse
	4	Sistem menampilkan *open file* dialog
	5	*Manager* memilih data transaksi
	6	Sistem menampilkan data transaksi
	7	*Manager* memilih simpan
	8	Sistem menyimpan data

Extension	Step	Branching Action
	2.1	Menampilkan notifikasi kesalahan terhadap masukan
	6.1	Menampilkan notifikasi gagal menyimpan data ke *database*

2. Skenario *Bworse*

Skenario *use case browse* menjelaskan langkah-langkah untuk melakukan pemilihan *file* transaksi yang akan disimpan ke *database*

Tabel 3. 47 Skenario *Use Case browse*

Use case Name	*Browse*	
Related Requirements	Requirement *A.1*	
Goal In Context	*Import* data transaksi Mini Market Barokah dan menyimpan ke dalam *database*	
Precondition	*Manager* menyiapkan data transaksi.	
Successful End Condition	Data transaksi yang di *import* oleh *manager* berhasil disimpan ke dalam *database*	
Failed End Condition	Data transaksi gagal disimpan ke dalam *database*	
Actors	*Manager*	
Triger	*Manager* memasukan data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*Manager* klik *browse*
	2	Sistem menampilkan open *file* dialog
	3	*Manager* memilih data transaksi
	4	Sistem menampilkan data transaksi
Extension	Step	Branching Action
	-	

3. Skenario Simpan Data

Skenario *use case* simpan data menjelaskan langkah-langkah untuk melakukan proses penyimpanan data ke dalam *database.*

Tabel 3. 48 Skenario *Use Case* simpan data

Use case Name	Simpan data	
Related Requirements	Requirement *A.1*	
Goal In Context	Sistem melakukan proses penyimpanan data ke dalam *database*	
Precondition	Data transaksi sudah tersimpan didalam *database*	
Successful End Condition	Sistem berhasil melakukan proses penyimpanan data.	
Failed End Condition	Sistem gagal untuk melakukan penyimpanan data.	
Actors	*Manager*	
Triger	*Manager* melakukan penyimpanan data	
Included Cases	Menyimpan data ke *database*	
Main Flow	Step	Action
	1	*Manager* klik simpan
	2	Sistem melakukan proses simpan data
	3	Sistem menyimpan data ke *database*
Extension	Step	Branching Action
	-	

.

4. Skenario *Use Case Preprocesing*

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

Tabel 3. 49 Requirement A.2

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

Tabel 3. 50 Skenario *Use Case Preprocesing*

Use case Name	*Preprocesing*	
Related Requirements	*Requirement A.1, Requirement A.2*	
Goal In Context	Sistem melakukan proses pembersihan data, pemilihan atribut, dan hitung *support count*.	
Precondition	Data transaksi sudah tersimpan didalam *database*	
Successful End Condition	Sistem berhasil melakukan proses pembersihan data, pemilihan atribut, dan hitung *support count*.	
Failed End Condition	Sistem gagal untuk melakukan proses pembersihan data, pemilihan atribut, dan hitung *support count*.	
Actors	*User*	
Triger	*User* melakukan *request* proses pembersihan data, pemilihan atribut, dan hitung *support count*.	
Included Cases	Pembersihan Data, Pemilihan Atribut, Dan Hitung *Support Count*.	
Main Flow	Step	Action
	1	*Manager* klik menu *preprocessing*
	2	Sistem menampilkan menu preprocessing
	3	*Manager* klk *preprocessing*
	4	Sistem melakukan pembersihan data *non food* dan tunggal

	5	Sistem menyimpan data yang sudsh dibersihkan ke *database*
	6	Sistem menampilkan data yang sudah dibersihkan
	7	Sistem melakukan pemilihan atribut no transaksi dan nama barang
	8	Sistem menympan data yang sudah dipilih ke *database*
	9	Sistem menampilkan data yang sudah dpilih atributnya
	10	Sistem melakukan hitung *support count*
	11	Menyimpan perhitungan *support count* ke *database*
	12	Menampilkan perhitungan *support count*
	13	Data berhasil disimpan
Extension	Step	Branching Action
	2.1	Sistem menampilkan pesan kesalahan dalam proses preprocessing

5. Skenario *Use Case* Proses Asosiasi

Skenario *use case* proses asosiasi menggambarkan langkah-langkah aksi *user* untuk melakukan proses memasukan nilai *minimum confidance* dan *minimum support*, Lihat *Rules*, dan Lihat Informasi Paket awal sekenario *use case* proses asosiasi dapat dilihat pada tabel 3.51

Tabel 3. 51 Requirement A.3

Requirement A.3
Sistem menyediakan menu asosiasi untuk memasukan *Minimum Support* dan *Minimum Confidence* untuk mengeliminasi barang apa saja yang tidak memenuhi nilai parameter yang diinputkan, Lihat *rules* untuk melihat *rule* apa saja yang dihasilkan, dan Lihat Informasi paket untuk melihat informasi pembentukan paket yang dihasilkan

Tabel 3. 52 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 kemudian akan ditampilkan dalam bentuk informasi paket barang	
Precondition	Data sudah melalui proses *preprocessing*	
Successful End Condition	Sistem berhasil mendapatkan *rule* yang kemudian akan ditampilkan dalam bentuk informasi paket barang	
Failed End Condition	Sistem gagal mendapatkan *rule* yang akan ditampilkan dalam bentuk informasi paket barang	
Actors	*User*	
Triger	*User* menginputkan nilai parameter berupa nilai *Minimum Support* dan *Minimum Confidence*	
Included Cases	Lihat *Rules*, Lihat Informasi Paket	
Main Flow	Step	Action
	1	*Manager* klik menu asosasi
	2	Sistem menampilkan menu asosiasi

Main Flow	Step	Action
	3	*Manager* memasukan nilai *conidance* dan *minimum support*
	4	*Manager* klik proses
	5	Sistem melakukan *generate rule*
	6	Sistem melakukan Eliminasi terhadap item yang kurang dari nilai *Minimum Confidence*
	7	Sistem mendapatkan hasil *rule* yang sesuai dengan *minimum confidence*
	8	Sistem menampilkan hasil *rule* dan menampilkan hasil informasi paket barang
Extension	Step	Branching Action
	3.1	*Minimun support* dan *minimum confidance* tidak boleh kosong

6. Skenario *Use Case* Pembentukan Paket

Skenario *use case* Pembentukan paket menggambarkan langkah-langkah aksi *user* untuk melakukan proses memasukan harga minimal dan maksimal , skenario *use case* pembentukan harga dapat dilihat pada tabel 3.53.

Tabel 3. 53 Skenario *Use Case* Pemaketan

***Requirement* A.3**
Sistem menyediakan pembentukan paket untuk memasukan minimal dan maksimal harga untuk membentuk paket barang, Lihat pembentukan paket barang apa saja yang dihasilkan

Tabel 3. 54 Skenario Use Case Pemaketan

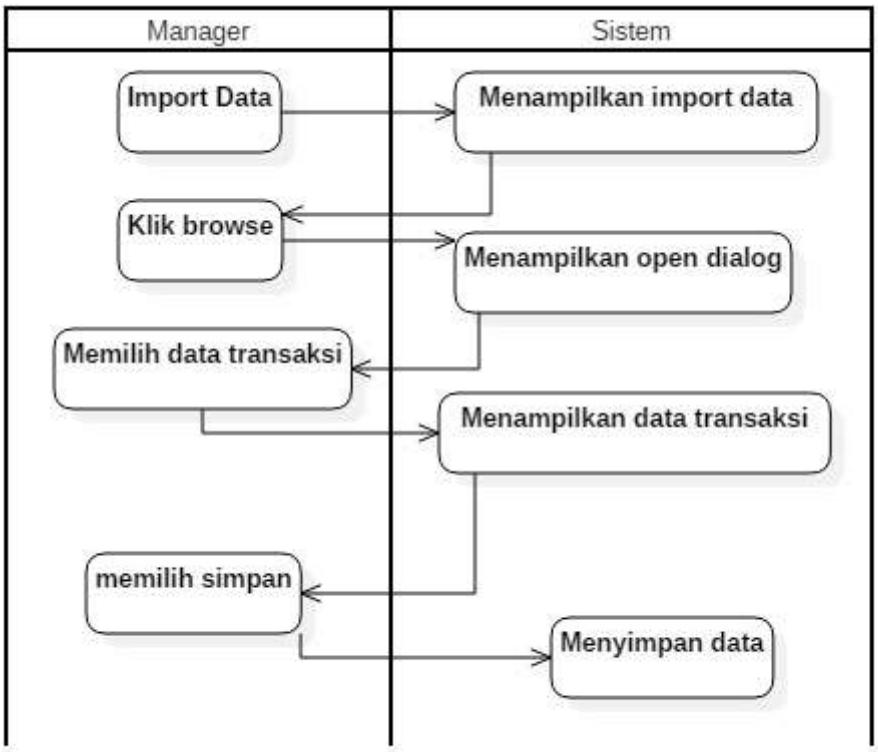
Use case Name	Pemaketan	
Goal In Context	Sistem menampilkan informasi paket barang hasil *filtering rule*	
Precondition	Sistem telah menghasilkan *rule*	
Successful End Condition	Sistem berhasil menampilkan informasi paket barang hasil *filtering rule*	
Failed End Condition	Sistem gagal menampilkan informasi paket barang hasil *filtering rule*	
Actors	*Manager* memasukan harga minimal dan maksimal	
Triger	*Manager*	
Main Flow	Step	Action
	1	*Manager* klik paket
	2	Sistem menampilkan menu paket
	3	*Manager* memasukan harga maksimal dan minimal
	4	*Manager* klik *filter*
	5	Sistem melakukan perhitungan harga
	6	Sistem menampilkan rule
Extension	Step	Branching Action
	1.1	Minimun harga dan maksimal harga tidak boleh kosong

3.1.4.3 **Diagram Activity**

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

1. *Activity Diagram Import* Data

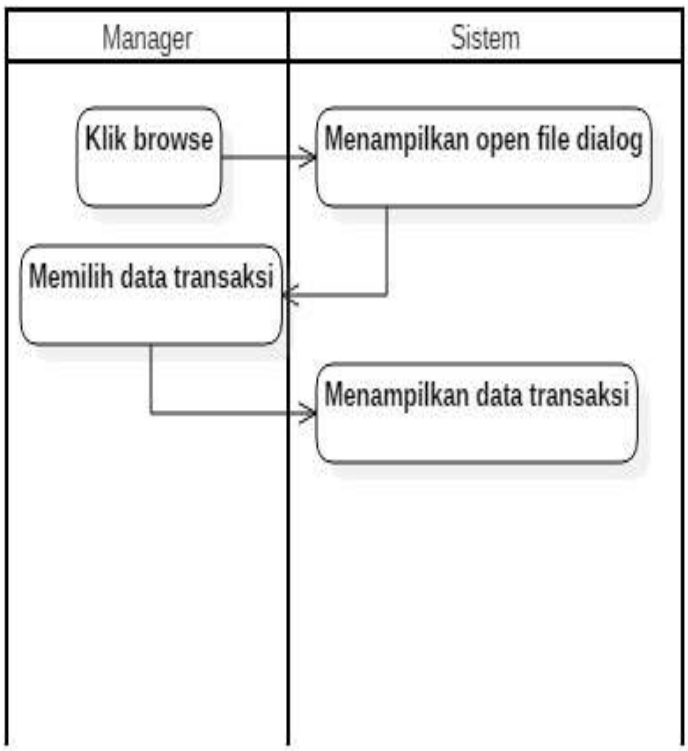
   Activity diagram import data menjelaskan aktivitas aktor dalam melakukan proses *import* data. Penjelasan *activity diagram import data* diacu pada Gambar 3.35.



Gambar 3. 35 *Activity Diagram Import Data*

2. *Activity Diagram browse*

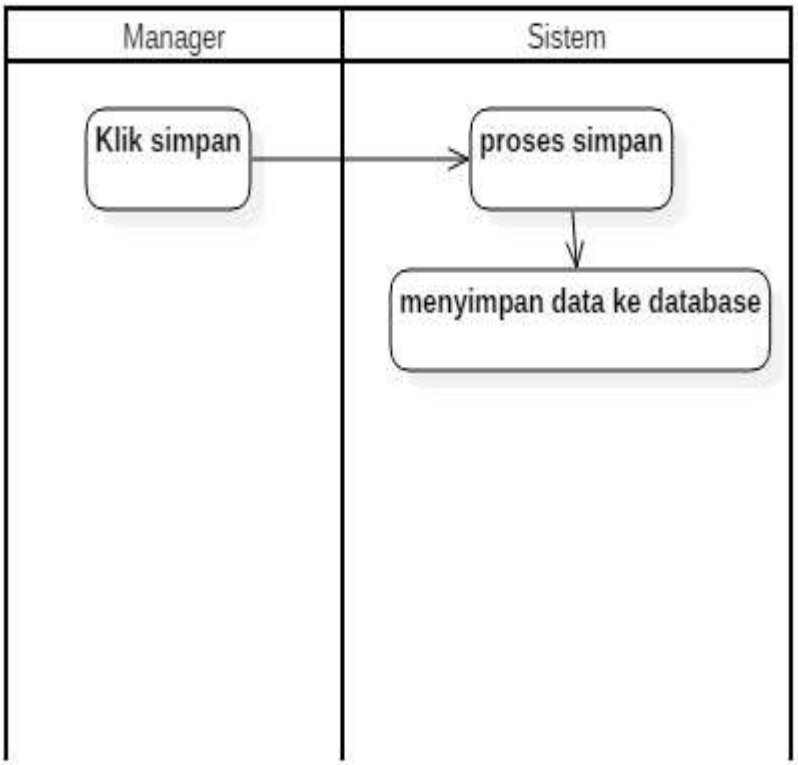
Activity diagram browse menjelaskan aktivitas sistem dalam melakukan proses pencarian data yang akan dipakai. *Activity diagram browse* dapat dilihat pada gambar 3.36.



Gambar 3. 36 *Activity Diagram Browse*

3. Diagram *Activity* Simpan Data

Diagram *activity* simpan data menjelaskan sistem dalam melakukan proses penyimpanan data ke *database*. *Activity diagram browse* dapat dilihat pada gambar 3.37.



Gambar 3. 37 *Activity Diagram* Simpan Data

4. *Activity Diagram Preprocesing*

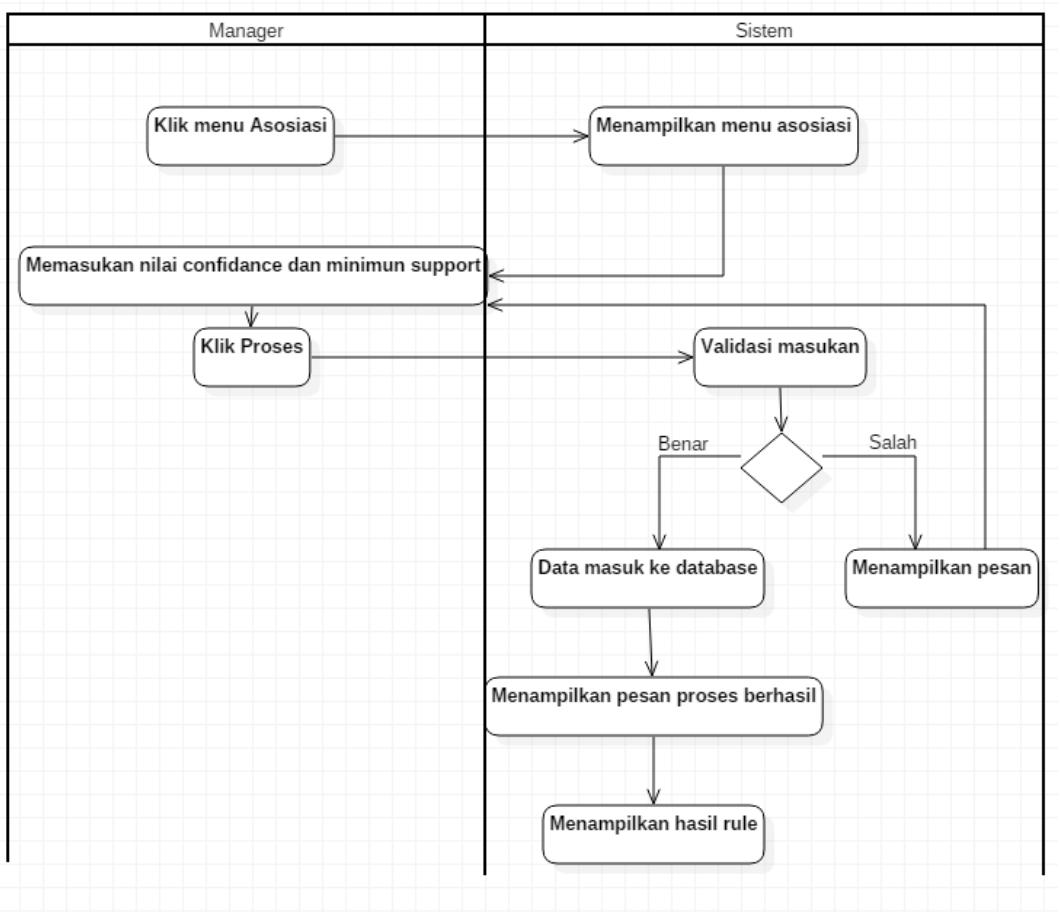
Activity diagram preprocessing menjelaskan aktivitas sistem dalam melakukan proses Pembersihan data dan Pemilihan Atribut . *Activity diagram preprocessing* dapat dilihat pada gambar 3.38.



Gambar 3. 38 *Activity Diagram Preprocessing*

5. *Activity Diagram* Asosiasi

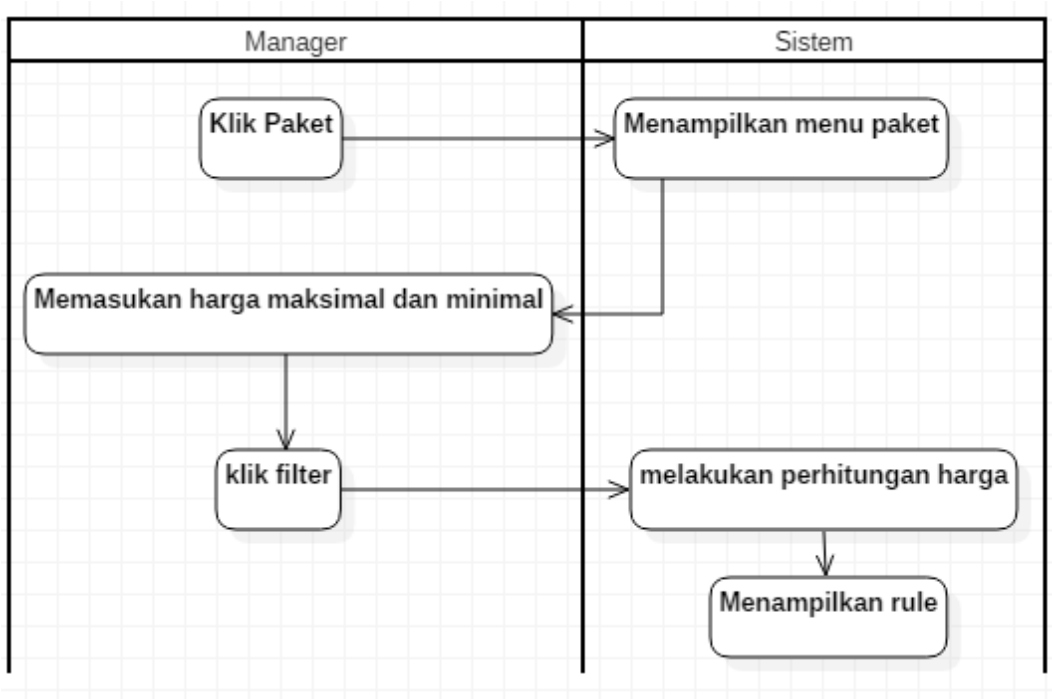
Activity diagram asosiasi menjelaskan aktivitas user dan sistem dalam melakukan proses memasukan nilai *confidance* dan *minimun support*.*Activity diagram* Asosiasi dapat dilihat pada gambar 3.39.



Gambar 3.39 *Activity Diagram* Asosiasi

6. *Activity diagram* pembentukan paket

Activity diagram pembentukan paket menjelaskan aktifivitas *user* dan sistem dalam menentukan minimal harga dan maksimal harga sesuai dengan yang dimaksud oleh *user*. *Activity diagram* lihat hasil pembentukan paket dapat dilihat pada gambar 3.40.



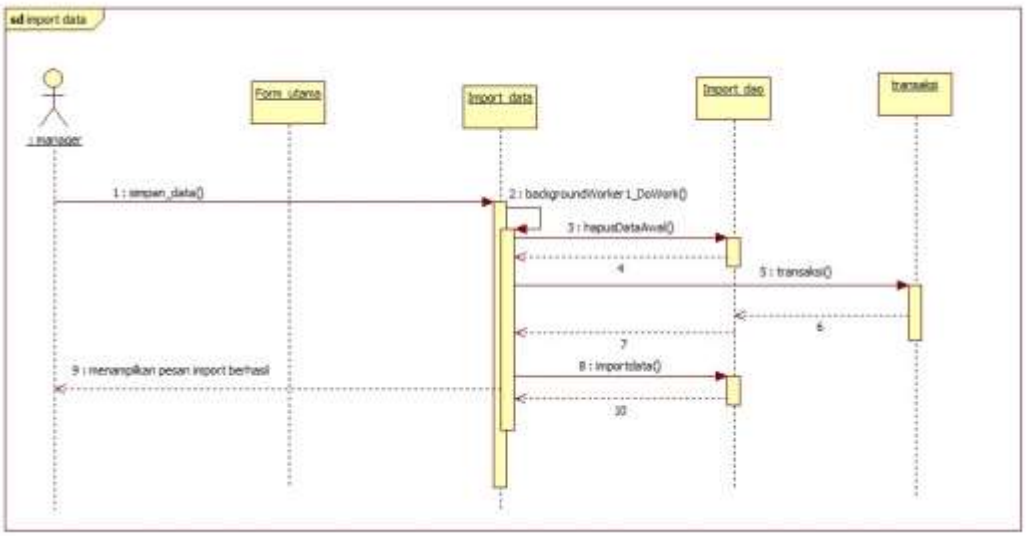
Gambar 3. 40 *Activity Diagram* Pemaketan

3.1.4.4 Diagram *Sequence*

Diagram sequensial menggambarkan interaksi objek pada *use case* dengan mendeksripsikan waktu hidup objek dan pesan yang dikirimkan dan di terima antar objek. Berikut adalah diagram squensial dari sistem yang akan dibangun :

1. *Sequence* Diagram Proses *Import Data*

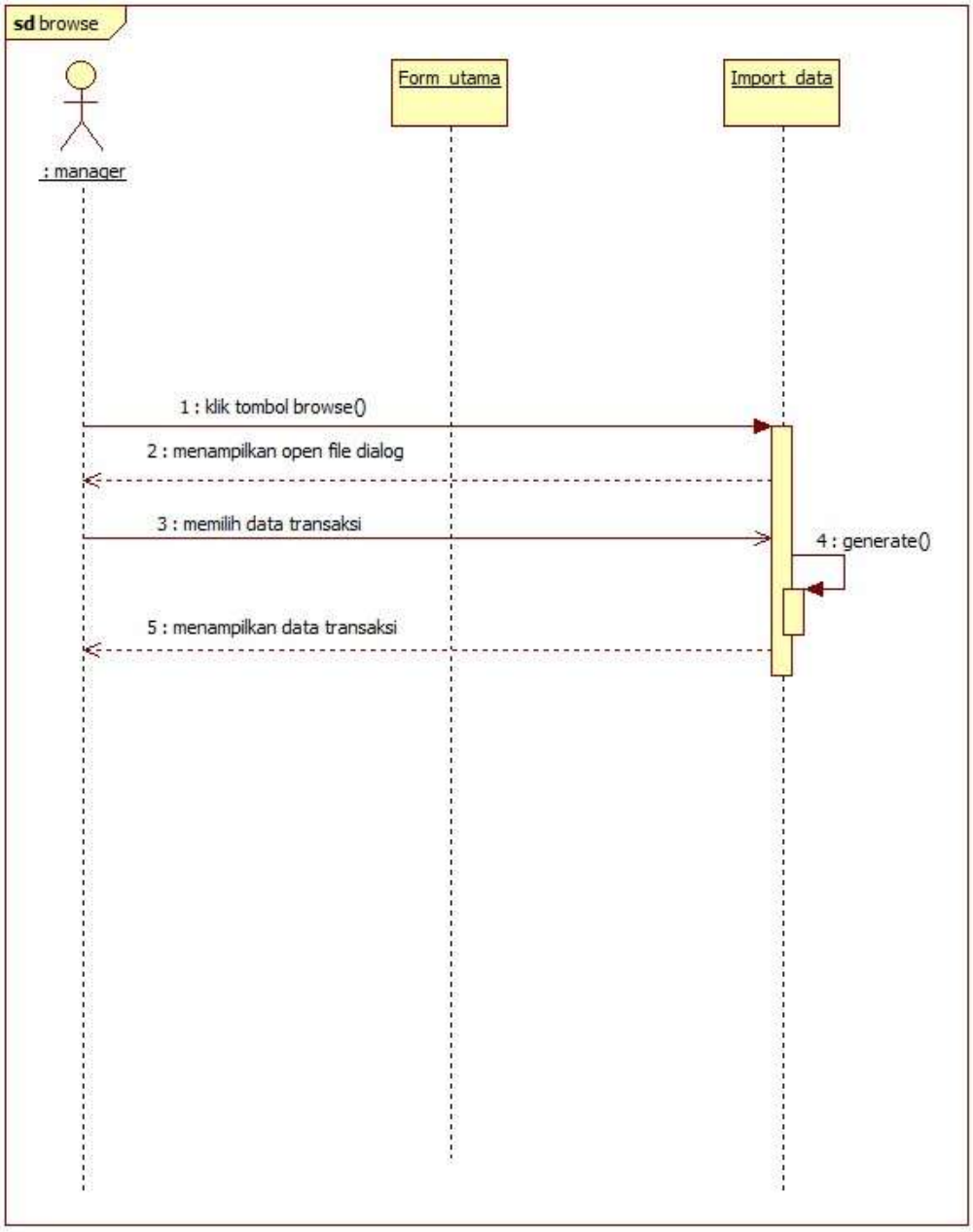
   Untuk *sequence diagram* pada proses import data dapat dilihat pada gambar 3.41.



Gambar 3. 41 *Squence Diagram Import Data*

2. *Sequence Diagram browse*

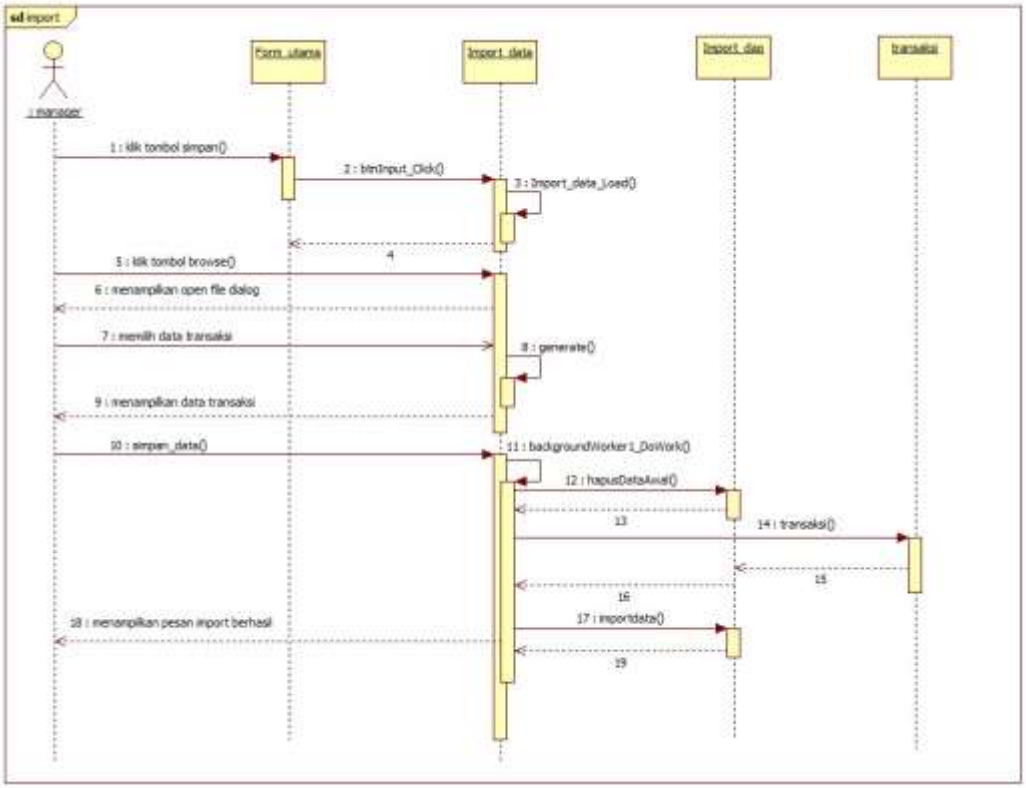
Untuk *sequence diagram* pada *browse* dapat dilihat pada gambar 3.42.



Gambar 3. 42 *Squence Diagram browse*

3. *Squence Diagram* Proses simpan data

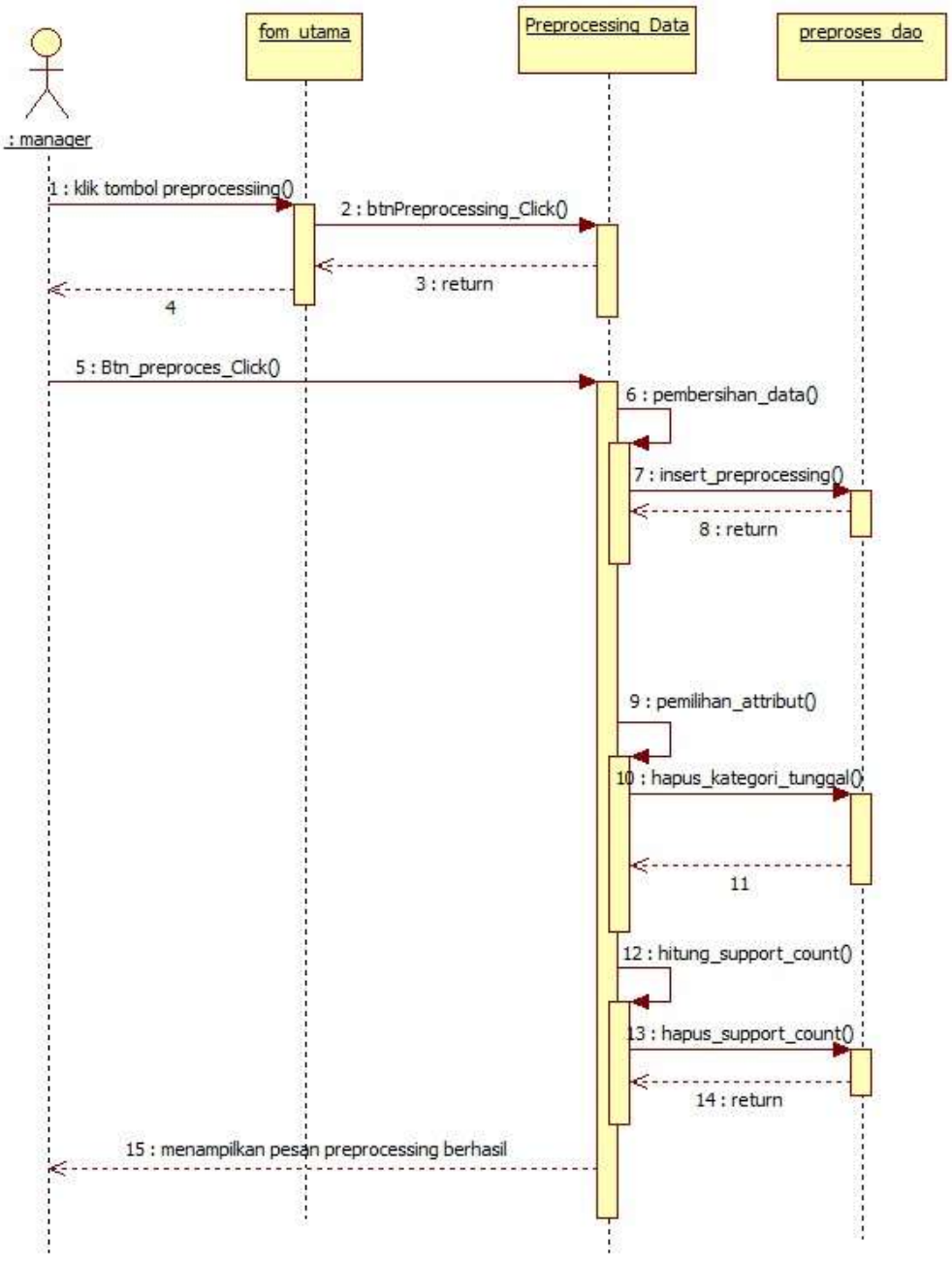
Untuk *sequence diagram* pada proses simpan data dapat dilihat pada gambar 3.43.



Gambar 3. 43 *Squence Diagram* simpan data

4. *Squence Diagram Preprocessing*

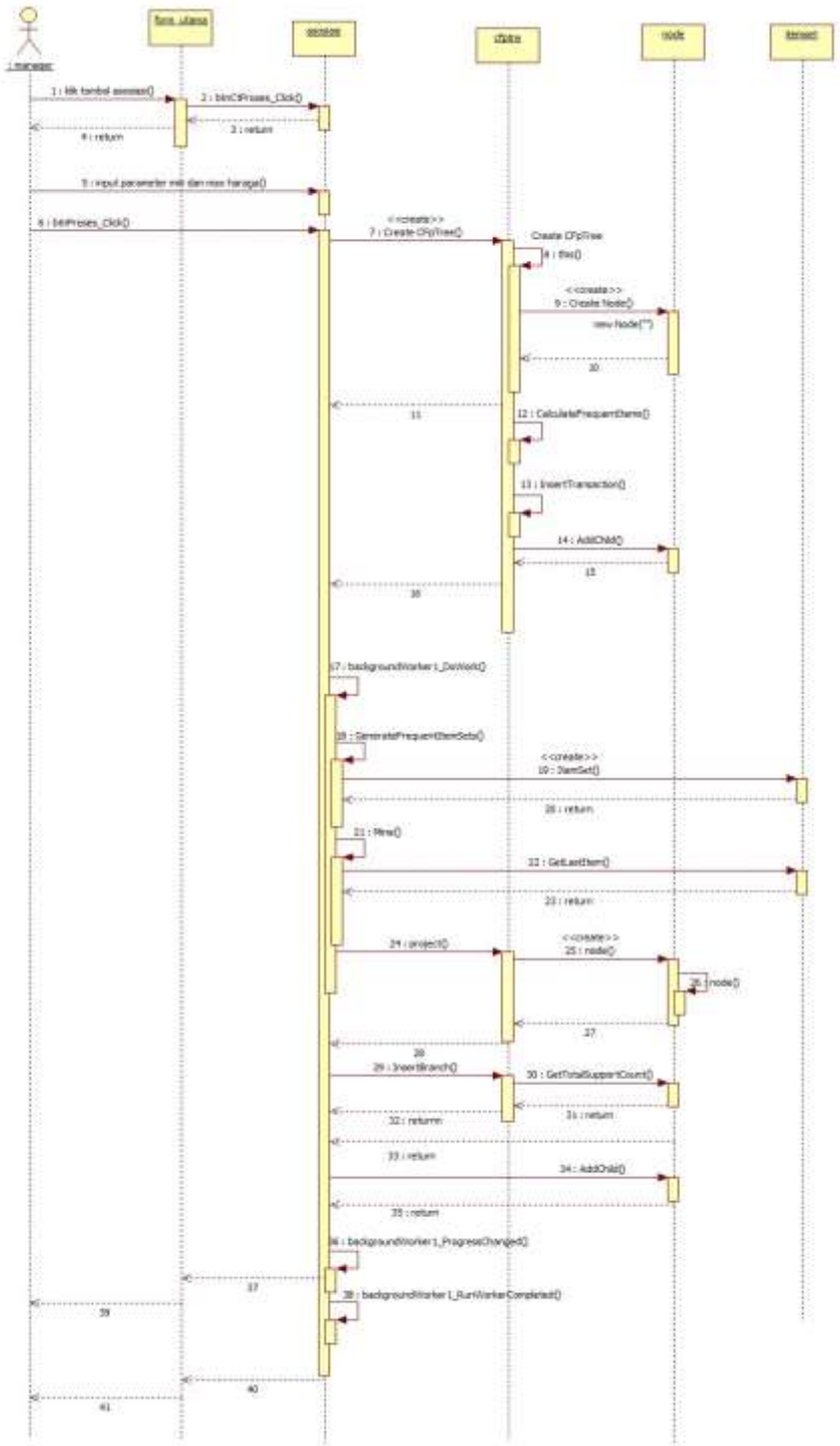
Untuk *sequence diagram* pada proses *preprocesing* dapat dilihat pada gambar 3.44.



Gambar 3. 44 *Squence preprocessing*

5. *Squence Diagram* Proses Asosiasi

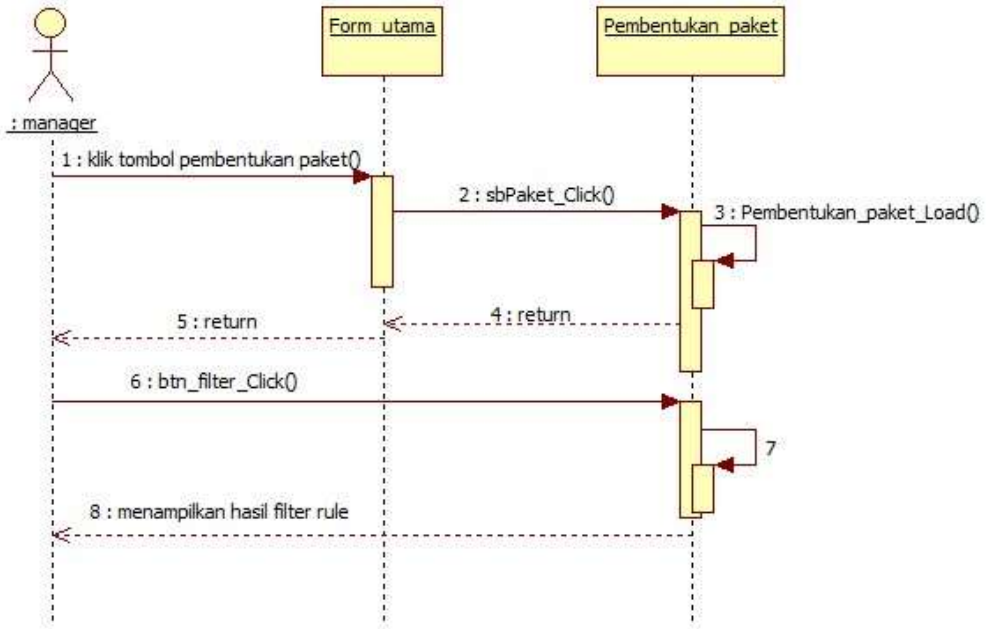
Untuk *sequence diagram* pada proses asosiasi dapat dilihat pada gambar 3.45.



Gambar 3. 45 *Squence Diagram* Asosiasi

6. *Squence Diagram* Pembentukan Paket

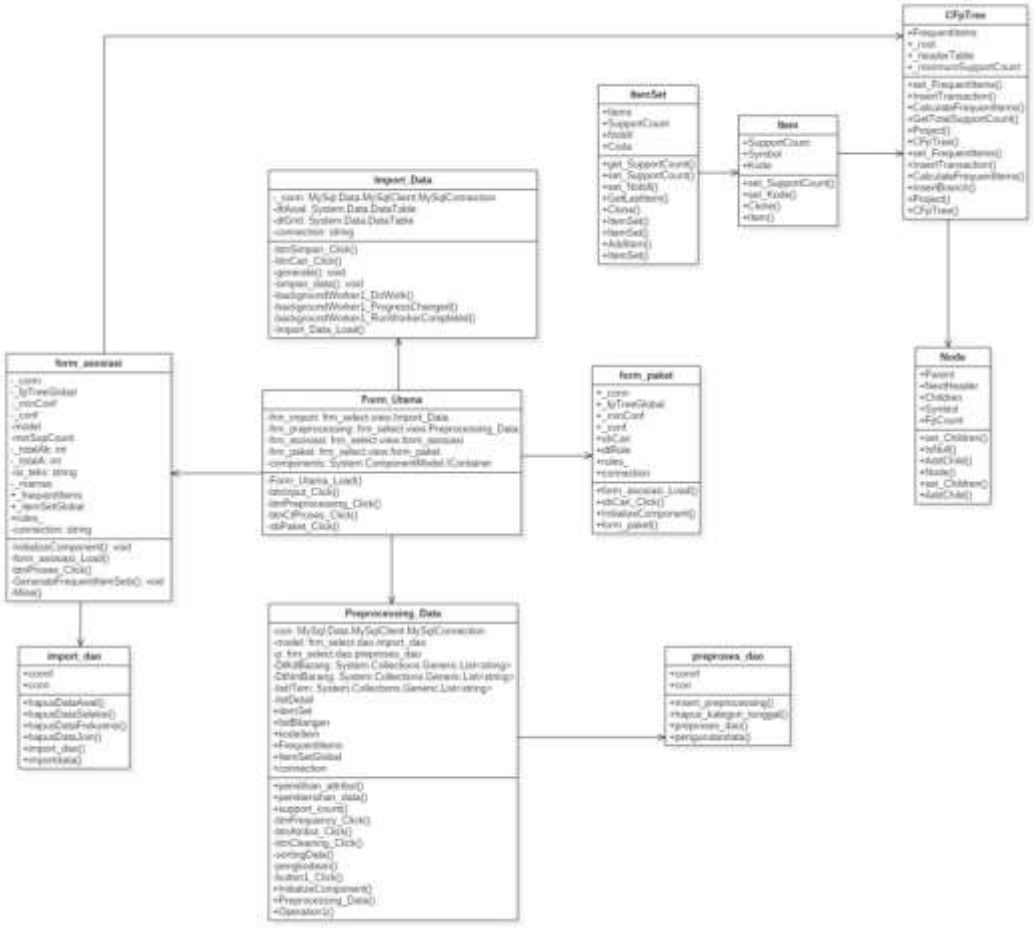
Untuk *sequence diagram* pada proses pembentukan paket dapat dilihat pada gambar 3.46.



Gambar 3. 46 *Squence Diagram* Pembentukan Paket

3.1.4.5 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Diagram Kelas dalam aplikasi yang akan dibangun ini digambarkan pada gambar 3.47.



Gambar 3. 47 *Class Diagram*

3.2 Perancangan Sistem

Tahap perancangan sistem merupakan tahap untuk melakukan perencanaan rancangan terhadap sistem yang akan dibangun, mulai dari perancangan basis data, antarmuka perangkat lunak, perancangan struktur menu, perancangan pesan, jaringan semantik dan perancangan prosedural. Dimana semua perancangan yang di buat tersebut harus saling berhubungan satu sama lain agar dapat menghasilkan suatu sistem yang baik.

3.2.1 Perancangan Kelas

Berikut ini adalah perancangan yang digunakan pada sistem yang akan dibangun pada tabel 3.55 di bawah:

Tabel 3. 55 Struktur Kelas Form_Utama

Class : Form_Utama		
Nama	Jenis	*Type*
frm_import	Atribut	*Private*
frm_preprocessing	Atribut	*Private*
frm_asosiasi	Atribut	*Private*
frm_paket	Atribut	*Private*
components	Atribut	*Private*
Form_Utama_Load	*Method*	*Private*
btnInput_Click	*Method*	*Private*
btnPreprocessing_Click	*Method*	*Private*
btnCtProses_Click	*Method*	*Private*
sbPaket_Click	*Method*	*Private*

Tabel 3. 56 Struktur Kelas Form_*Importdata*

Class : Import_data		
Nama	Jenis	*Type*
_conn	Atribut	*Private*
dtAwal	Atribut	*Private*
dtGrid	Atribut	*Private*
Connection	Atribut	*Private*
btnSimpan_Click	*Method*	*Private*
btnCari_Click	*Method*	*Private*
Generate	*Method*	*Private*
simpan_data	*Method*	*Private*
backgroundWorker1_DoWork	*Method*	*Private*
backgroundWorker1_ProgressChanged	*Method*	*Private*
backgroundWorker1_RunWorkerCompleted	*Method*	*Private*
Import_Data_Load	*Method*	*Private*

Tabel 3. 57 Struktur Kelas *Import*_dao

Class : Import_dao		
Nama	Jenis	*Type*
Connf	Atribut	*Private*
Conn	Atribut	*Private*
importdata	*Method*	*Public*
hapusDataAwal	*Method*	*Public*
hapusDataSeleksi	*Method*	*Public*
hapusDataFrekuensi	*Method*	*Public*
hapusDataJoin	*Method*	*Public*
import_dao	*Method*	*Public*
importdata	*Method*	*Public*

Tabel 3. 58 Struktur Kelas Prepocessing_data

Class : preprocessing		
Nama	Jenis	*Type*
Con	Atribut	*Public*
Model	Atribut	*Private*
P	Atribut	*Private*
DtKdBarang	Atribut	*Private*
DtNmBarang	Atribut	*Private*
listITem	Atribut	*Private*
List	Atribut	*Private*
listDetail	Atribut	*Private*
itemSet	Atribut	*Private*
listBilangan	Atribut	*Private*
kodeItem	Atribut	*Private*
FrequentItems	Atribut	*Public*
ItemSetGlobal	Atribut	*Public*
connection	Atribut	*Private*
pemilihan_attribut	*Method*	*Public*
pembersihan_data	*Method*	*Public*
*support*_count	*Method*	*Public*
btnFrequency_Click	*Method*	*Private*
btnAtribut_Click	*Method*	*Private*
btnCleaning_Click	*Method*	*Private*
sortingData	*Method*	*Private*
pengkodean	*Method*	*Private*
button1_Click	*Method*	*Private*
InitializeComponent	*Method*	*Private*
Preprocessing_Data	*Method*	*Public*
pengurutandata	*Method*	*Public*

Tabel 3. 59 Struktur Kelas form_asosiasi

Class : form_asosiasi		
Nama	Jenis	*Type*
_conn	Atribut	*Private*
_fpTreeGlobal	Atribut	*Private*
_minConf	Atribut	*Private*
_conf	Atribut	*Private*
Model	Atribut	*Private*
minSupCount	Atribut	*Private*
_totalAb	Atribut	*Private*
_totalA	Atribut	*Private*
isi_teks	Atribut	*Private*
_mamas	Atribut	*Private*
_frequentItems	Atribut	*Public*
_itemSetGlobal	Atribut	*Public*
rules_	Atribut	*Public*
connection	Atribut	*Private*
InitializeComponent	*Method*	*Private*
form_asosiasi_Load	*Method*	*Private*
btnProses_Click	*Method*	*Private*
GenerateFrequentItemSets	*Method*	*Private*
Mine	*Method*	*Private*

Tabel 3. 60 Struktur Kelas from_paket

Class : form_paket		
Nama	Jenis	*Type*
_conn	Atribut	*Private*
_fpTreeGlobal	Atribut	*Private*
_minConf	Atribut	*Private*
_conf	Atribut	*Private*
sbCari	Atribut	*Private*
dtRule	Atribut	*Public*
rules_	Atribut	*Public*
connection	Atribut	*Private*
form_asosiasi_Load	*Method*	*Private*
sbCari_Click	*Method*	*Private*
InitializeComponent	*Method*	*Private*
form_paket	*Method*	*Public*

Tabel 3. 61 Struktur Kelas *Cfptree*

Class : CfpTree		
Nama	Jenis	*Type*
FrequentItems	Atribut	*Private*
_root	Atribut	*Private*
_headerTable	Atribut	*Private*
_minimum*Support*Count	Atribut	*Private*
set_FrequentItems	*Method*	*Private*
InsertTransaction	*Method*	*Private*
CalculateFrequentItems	*Method*	*Private*
InsertBranch	*Method*	*Private*
GetTotal*Support*Count	*Method*	*Private*
Project	*Method*	*Public*
CFpTree	*Method*	*Public*
set_FrequentItems	*Method*	*Private*
InsertTransaction	*Method*	*Private*
CalculateFrequentItems	*Method*	*Private*
InsertBranch	*Method*	*Private*
Project	*Method*	*Public*
CFpTree	*Method*	*Public*

Tabel 3.62. Struktur Kelas Node

Class : Node		
Nama	Jenis	*Type*
Parent	Atribut	*Private*
NextHeader	Atribut	*Private*
Children	Atribut	*Public*
Symbol	Atribut	*Public*
FpCount	Atribut	*Public*
set_Children	*Method*	*Private*
IsNull	*Method*	*Public*
AddChild	*Method*	*Public*
Node	*Method*	*Public*
set_Children	*Method*	*Private*
AddChild	*Method*	*Public*

Tabel 3. 63 Struktur Kelas item

Class : item		
Nama	Jenis	*Type*
*Support*Count	Atribut	*Public*
Symbol	Atribut	*Public*
Kode	Atribut	*Public*
set_*Support*Count	*Method*	*Private*
set_Kode	*Method*	*Private*
Clone	*Method*	*Public*
Item	*Method*	*Public*

Tabel 3. 64 Struktur Kelas *Itemset*

Class : Itemset		
Nama	Jenis	*Type*
Items	Atribut	*Private*
*Support*Count	Atribut	*Private*
Nobill	Atribut	*Public*
Coda	Atribut	*Public*
AddItem	*Method*	*Public*
get_*Support*Count	*Method*	*Private*
set_*Support*Count	*Method*	*Private*
set_Nobill	*Method*	*Private*
GetLastItem	*Method*	*Public*
Clone	*Method*	*Public*
ItemSet	*Method*	*Public*
ItemSet	*Method*	*Public*
AddItem	*Method*	*Public*
ItemSet	*Method*	*Public*

3.2.2 Perancangan Data

3.2.2.1 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. Data Transaksi
2. Data *Preprocessing*
3. Data *Support Count*
4. Data *Rules*

3.2.2.2 Struktur Tabel

Struktur tabel merupakan gambaran *detail* tabel yang berisi *field*, tipe data, panjang data, dan keterangan lainnya. Adapun tabel-tabel yang digunakan dalam perangkat lunak yang dibangun yang diacu dari tabel 3.67 sampai tabel 3.70.

1. Tabel Transaksi

   Tabel transaksi merupakan tabel yang digunakan untuk menyimpan data transaksi hasil import dari *file* excel. Dapat dilihat pada tabel 3.65 di bawah:

Tabel 3. 65 Data Awal

Nama Field	Tipe Data	Panjang	Keterangan
Id	*Int*	11	*Primary Key*
Tanggal	*Varchar*	20	
no_transaksi	*Varchar*	20	
kode_barang	*Varchar*	30	
nama_barang	*Varchar*	30	
Kategori	*Varchar*	20	
jumlah	*Int*	11	
Harga	*double*		
Total	*Varchar*	50	

2. Tabel *Preprocessing*

Tabel *preprocessing* adalah tabel yang digunakan untuk menyimpan data hasil *preprocessing*. Dapat dilihat pada tabel 3.66 di bawah:

Tabel 3. 66 Data Preprocessing

Nama Field	Tipe Data	Panjang	Keterangan
nomor_transaksi	*Varchar*	30	
kode_barang	*Varchar*	30	
nama_barang	*Varchar*	30	
Id	*Int*	11	Foreign Key dari tabel transaksi

3. Tabel *Support Count*

Tabel *support count* adalah tabel yang digunakan untuk menyimpan data hasil penghitungan kemunculuan banyaknya masing-masing jenis barang. Dapat dilihat pada tabel 3.67 di bawah:

Tabel 3. 67 Data *Support* Count

Nama Field	Tipe Data	Panjang	Keterangan
Kode_barang	*Varchar*	30	
Support count	*Int*	5	
Id	*Int*	11	Foreign Key dari tabel transaksi

4. Tabel *Rules*

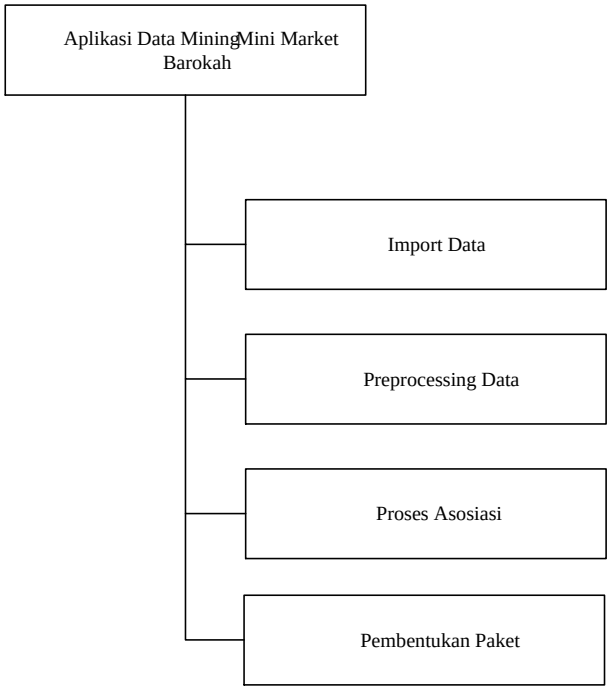
Tabel *rules* adalah tabel yang digunakan untuk menyimpan hasil pencarian *Frequent Itemset.* Dapat dilihat pada tabel 3.68 di bawah:

Tabel 3. 68 Data Pembentukan Rules

Nama Field	Tipe Data	Panjang	Keterangan
Rule	*Varchar*	200	
Support	*Int*	11	
Confidence	*Float*		
Id	*Int*	11	Foreign Key dari tabel *support* count

3.2.3 Perancangan Stuktur Menu

Dalam pembangunan perangkat lunak diperlukan sebuah struktur menu untuk menunjukkan keterkaitan dan tingkatan antara menu utama dengan sub menu, sehingga lebih mudah dipahami. Berikut adalah struktur menu dari perangkat lunak yang akan dibangun yang diacu pada gambar 3.48.



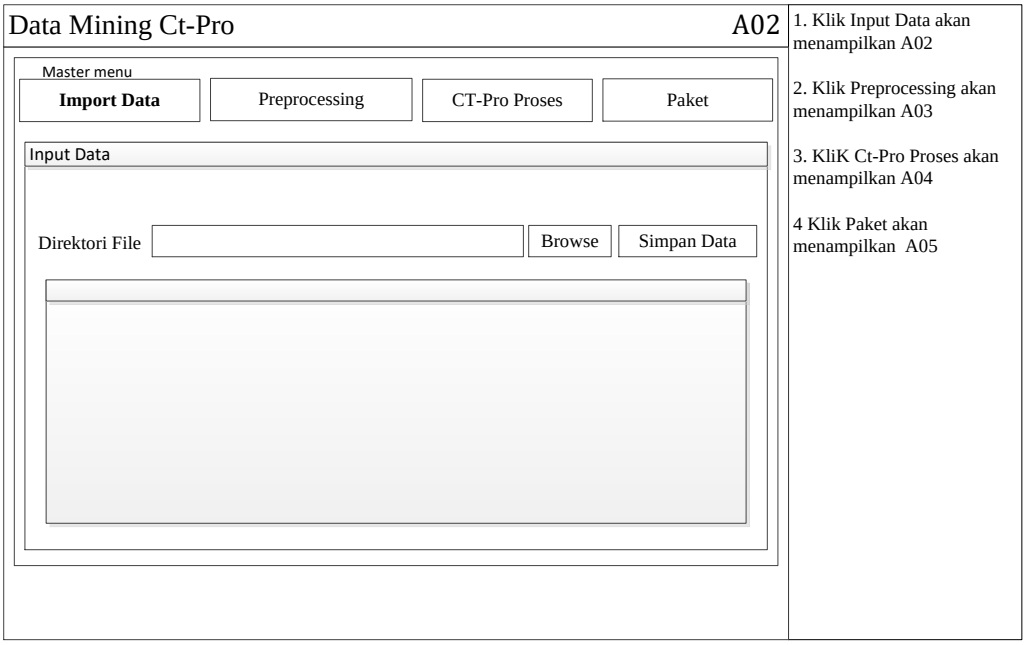
Gambar 3. 48 Struktur Menu

3.2.4 Perancangan Antar Muka

Perancangan antarmuka atau *interface* merupakan tampilan dari sebuah program atau aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana dialog antara program dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang sesuai dengan kebutuhan dan karakteristik *user*. Perancangan *interface* untuk aplikasi *Data mining* Mini Market Barokah adalah sebagai berikut:

1. Perancangan Antar Muka *Import* Data

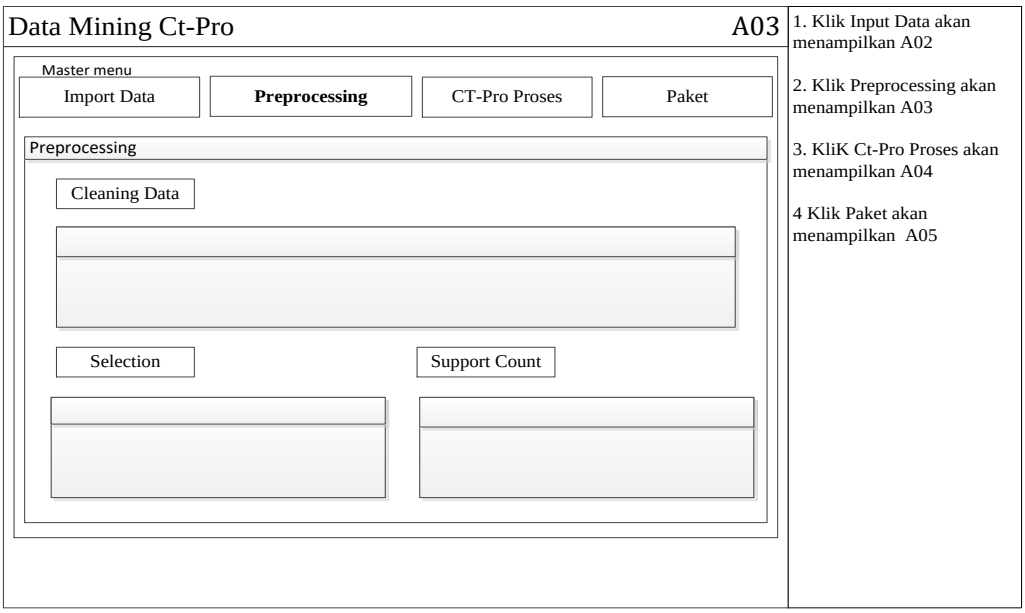
Perancangan Antar Muka *Import* Data dapat dilihat pada gambar 3.49.



Gambar 3. 49 Antar Muka Import Data

2. Perancangan Antar Muka *Preprocessing* Data

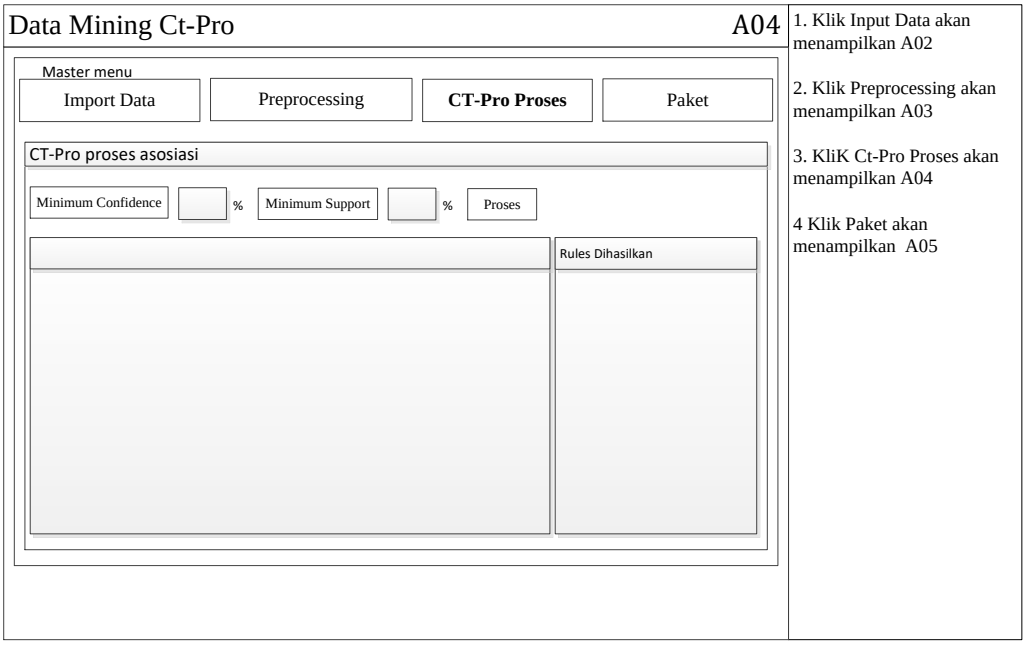
Perancangan Antar Muka *Preprocessing* Data dapat dilihat pada gambar 3.50.



Gambar 3. 50 Antar Muka *Preprocessing* Data

3. Perancangan Antar Muka Proses Asosiasi

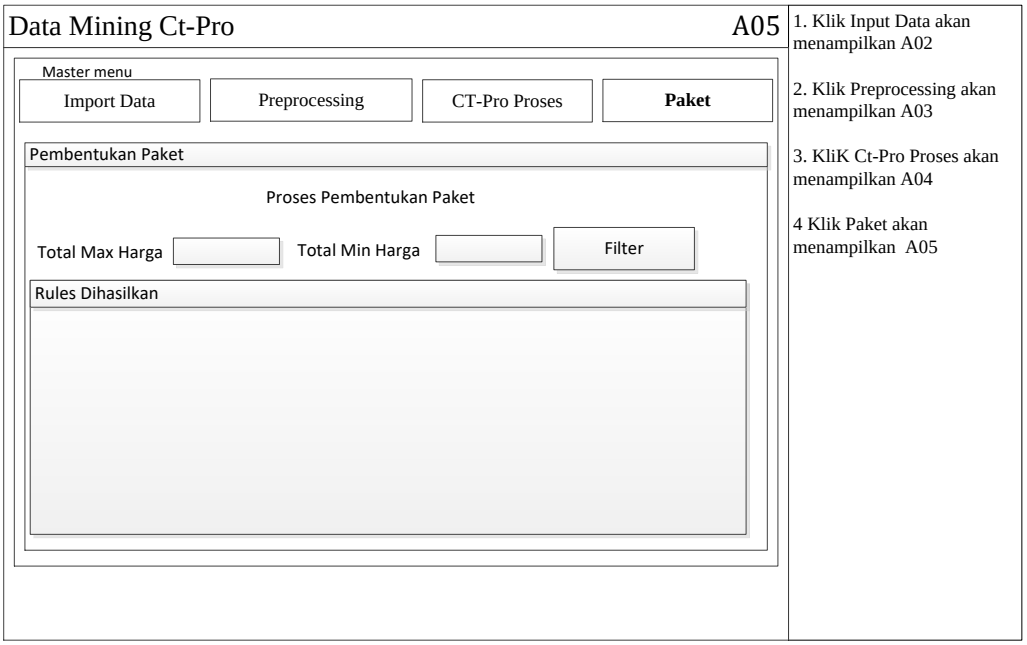
   Perancangan Antar Muka Proses Asosiasi dapat dilihat pada gambar 3.51



Gambar 3. 51 Antar Muka Proses Asosiasi

4. Perancangan Antar Muka Pembentukan Paket

   Perancangan Antar Muka Pembentukan Paket dilihat pada gambar 3.52



Gambar 3. 52 Antar Muka Pembuatan Paket

3.2.5 Perancangan Pesan

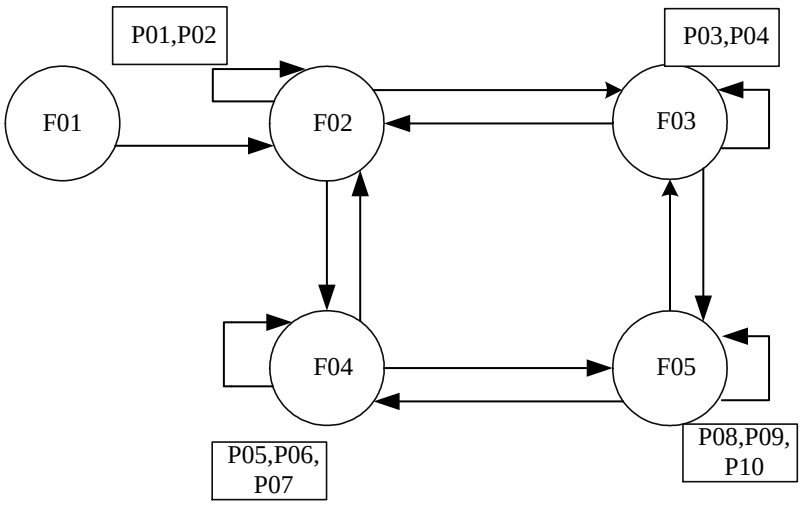
Adapun perancangan pesan dapat dilihat pada gambar 3.53 di bawah ini:



Gambar 3. 53 Antar Muka Perancangan Pesan

3.2.6 Perancangan Semantik

Perancangan Semantik dapat dilihat pada gambar 3.54.



Gambar 3. 54 Perancangan Semantik