BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem adalah penguraian dari suatu informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Analisis Sistem ini bertujuan untuk menjelaskan permasalahan yang muncul pada saat pembangunan sistem, hal ini bertujuan untuk membantu ketika proses perancangan sistem berlangsung. Tugas utama analisis sistem dalam tahap ini adalah menemukan kelemahan-kelemahan dari sistem yang berjalan sehingga dapat diusulkan perbaikannya. Dalam analisis sistem ini meliputi beberapa bagian, yaitu:

1. Analisis Masalah
2. Analisis Penerapan Metode *CRISP-DM*
3. Analisis Spesifikasi Kebutuhan Perangkat Lunak

3.1.1 Analisis Masalah

Berdasarkan hasil pengamatan, diketahui bahwa analisis masalahnya adalah sebagai berikut:

1. Mini Market HD Mart mengalami masalah dalam penentuan paket barang dikarenakan banyaknya barang yang harus dipertimbangkan, sehingga isi dari paket barang tersebut tidak sesuai dengan kebutuhan konsumen. Hal ini mengakibatkan pihak Mini Market HD Mart mengalami kerugian.
2. Dalam menentukan barang-barang apa saja yang akan dijual secara paket, Mini Market mengandalkan data barang yang banyak terjual, sehingga belum cukup dijadikan sebagai dasar pengambilan keputusan untuk membuat sebuah paket, karena barang yang terjual belum tentu efektif bila dijual secara bersamaan sebagai paket barang
3. Belum adanya informasi penentuan paket barang yang dapat ditawarkan kepada konsumen.

Berdasarkan hasil dari analisis masalah di atas, maka perlu dilakukannya penerapan Data *Mining* pada data transaksi penjualan menggunakan metode *association rule*s.

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 (CRISPDM).*

3.1.2.1 Pemahaman Bisnis

Tahapan pemahaman bisnis berfokus pada pemahaman tujuan kebutuhan berdasarkan penilaian bisnis. Kemudian pemahaman tersebut diubah menjadi sebuah rencana awal *Data Mining* yang dirancang untuk mencapai tujuan.

A. Tujuan Bisnis

Menghasilkan keuntungan dengan menjual barang dalam jumlah banyak dengan memanfaatkan penjual paket barang.

B. Menilai situasi

Menilai situasi merupakan tahapan untuk menjelaskan sumber daya yang tersedia di Mini Market HD Mart. Berikut Merupakan sumber daya yang ada pada mini Market HD Mart

a. *Inventory of resource*
   1. Sumber Daya Perangkat Keras
      a) Terdapat Alat penunjang transaksi yaitu *scanner barcode, printer thermal* dan *cashdrawer* (laci Kasir)
      b) Komputer
      c) Printer inject
   2. Sumber daya data
      a) Sumber daya data yang di Mini Market HD Mart Meliputi data transaksi, data stok barang, data pegawai, data supplier
      b) Sumber daya data di Mini Market ini bersifat internal yaitu tersimpan di *database* Mini Market HD Mart berupa data *sql*.

3. Sumber daya personil
    a) Pemilik Membuat perencanaan, strategi dan kebijakan yang menyangkut operasi Mini Market HD Mart.
    b) Pegawai kasir yang bertugas melakukan transaksi pembeli, mencetak nota dan membuat laporan penjualan yang akan di serahkan ke pemilik.
    c) Pegawai gudang yang bertugas mencatat barang masuk dan memeriksa stok barang.
    d) Admin keuangan bertanggung jawab terhadap hal-hal yang menyangkut keuangan

b. *Requirements, assumption and constraints*

a. Kebutuhan
    a) Data yang digunakan dalam penelitian ini adalah data transaksi periode bulan Januari 2016

b. Asumsi
    a) Data transaksi yang digunakan merupakan hasil exspor dari aplikasi kasir di Mini Market HD Mart dengan format *MS Excel(xlsx).*

c. Batasan
    a) Hasil dari *Data Mining* ini berupa *rule* sebagai acuan untuk membuat paket barang.
    b) Hasil dari *generate rule* akan dilakukan *filtering rule,* yaitu menyeleksi beberapa *rule* yang tidak sesuai dengan batasan kriteria bisnis Mini Market HD Mart dalam membentuk paket barang.
    c) Batasan untuk pembentukan paket berupa penentuan harga minimal dan maksimal yang dapat ditentukan oleh pengguna.

C. Penentuan Sasaran *Data Mining*

Mengetahui keterkaitan barang yang sering dibeli oleh konsumen sehingga dapat dijadikan landasan untuk merekomendasikan barang mana saja yang dapat dijadikan paket barang.

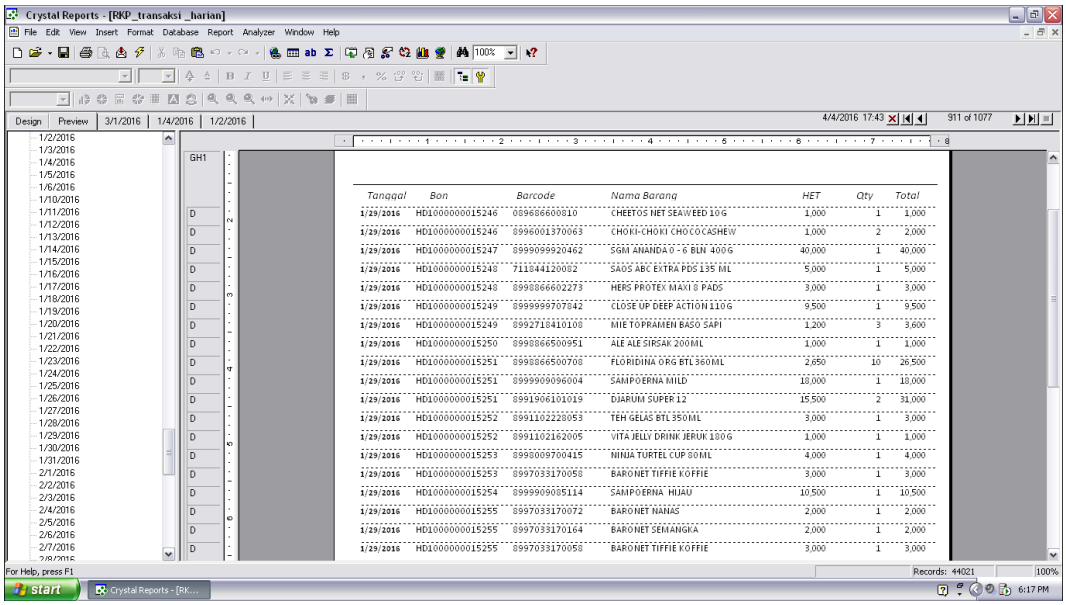
3.1.2.2 Pemahaman Data

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.

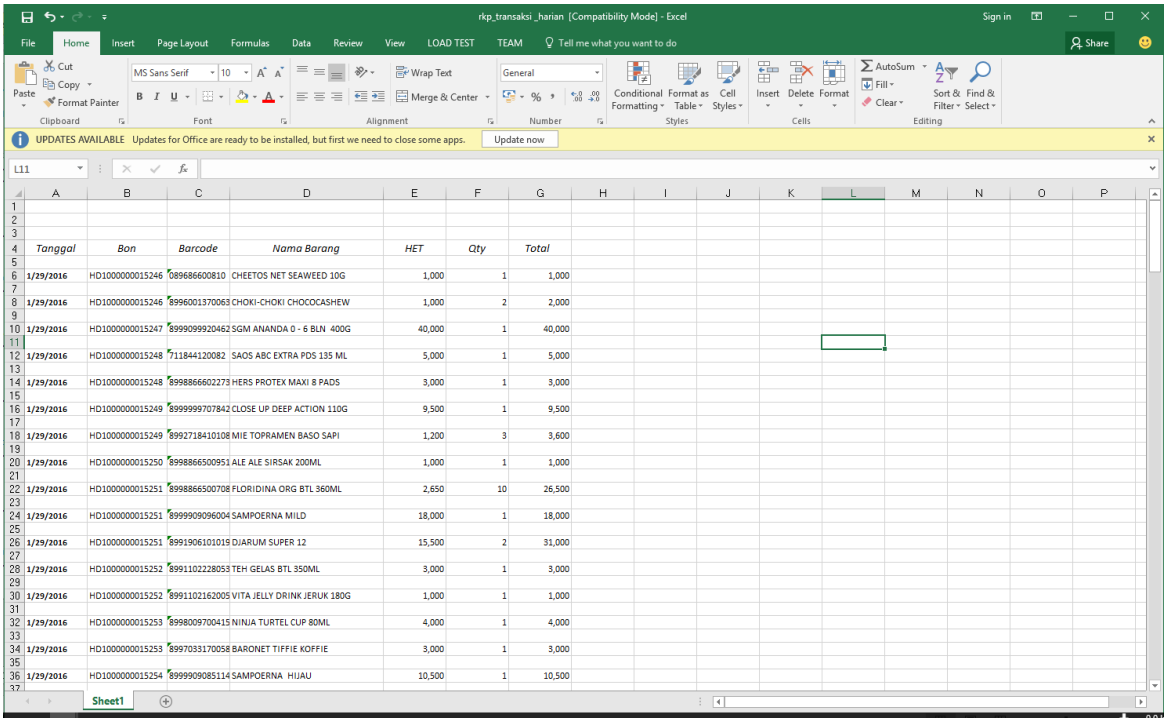
1. Mengumpulkan Data Awal

Data yang digunakan dalam penelitian ini yakni data transaksi penjualan barang Mini Market HD Mart. Berikut merupakan sumber data dalam bentuk *Crystal report* yang dapat di lihat pada gambar 3.1 di bawah:



Gambar 3. 1 Sumber Data Transaksi Mini Market HD Mart Dalam Bentuk *Ms Excel*

Data yang di hasilkan dari *Crystal report* tersebut kemudian di ekspor ke dalam bentuk *MS Excel(.xlsx)* berikut merupakan hasil ekspor dari *Crystal repot* yang dapat di lihat pada gambar 3.2 di bawah ini:



Tanggal	Bon	Barcode	Nama Barang	HET	Qty	Total
1/29/2016	HD1000000015246	089686600810	CHEETOS NET SEAWEED 10G	1,000	1	1,000
1/29/2016	HD1000000015246	8996001370063	CHOKI-CHOKI CHOCOCASHEW	1,000	2	2,000
1/29/2016	HD1000000015247	8999099920462	SGM ANANDA 0 - 6 BLN 400G	40,000	1	40,000
1/29/2016	HD1000000015248	711844120082	SAOS ABC EXTRA PDS 135 ML	5,000	1	5,000
1/29/2016	HD1000000015248	8998866602273	HERS PROTEX MAXI 8 PADS	3,000	1	3,000
1/29/2016	HD1000000015249	8999999707842	CLOSE UP DEEP ACTION 110G	9,500	1	9,500
1/29/2016	HD1000000015249	8992718410108	MIE TOPRAMEN BASO SAPI	1,200	3	3,600
1/29/2016	HD1000000015250	8998866500951	ALE ALE SIRSAK 200ML	1,000	1	1,000
1/29/2016	HD1000000015251	8998866500708	FLORIDINA ORG BTL 360ML	2,650	10	26,500
1/29/2016	HD1000000015251	8999909096004	SAMPOERNA MILD	18,000	1	18,000
1/29/2016	HD1000000015251	8991906101019	DJARUM SUPER 12	15,500	2	31,000
1/29/2016	HD1000000015252	8991102228053	TEH GELAS BTL 350ML	3,000	1	3,000
1/29/2016	HD1000000015252	8991102162005	VITA JELLY DRINK JERUK 180G	1,000	1	1,000
1/29/2016	HD1000000015253	8998009700415	NINJA TURTEL CUP 80ML	4,000	1	4,000
1/29/2016	HD1000000015253	8997033170058	BARONET TIFFIE KOFFIE	3,000	1	3,000
1/29/2016	HD1000000015254	8999909085114	SAMPOERNA HIJAU	10,500	1	10,500

Gambar 3. 2 Hasil Ekspor Data *Crystal report* dalam bentuk *Ms Excel (.xlsx)*

2. Mendeskripsikan Data

Tahap kedua dalam pemahaman data adalah mendeskripsikan data dengan tujuan memahami data yang didapatkan dari hasil pengumpulan data awal. Berikut merupakan penjelasan dari masing-masing atribut yang terdapat pada tabel data transaksi yang dapat dilihat pada tabel 3.1 di bawah ini:

Tabel 3. 1 Struktur Data Transaksi

Keterangan		
Fungsi	Untuk mengetahui jenis barang yang paling sering terjual oleh *customer*	
Format	*Microsoft Excel (.xlsx)*	
	Tanggal	Tanggal proses pembelian
	Bon	Nomor pembelian
	Barcode	Kode dari setiap barang
	Nama Barang	Nama setiap barang
	Het	Harga satuan barang

Keterangan		
	Qty	Banyaknya barang yang dibeli
	Total	Harga total dari pembelian tiap barang

3. Verifikasi Kualitas Data

Verifikasi kualitas data merupakan tahap di mana dilakukan pengecekan apakah data tersebut memiliki *noise* atau tidaknya. *Noise* di sini diartikan data transaksi yang hanya memiliki 1 barang dalam sekali pembelian, maka dari itu untuk menghilangkan *noise* pada data transaksi akan dilakukan proses pembersihan data pada tahap selanjutnya dalam tahapan *data preparation*.

3.1.2.3 Persiapan Data (*Data preparation*)

Persiapan data dilakukan dengan tujuan untuk menghasilkan sebuah data set baru yang akan digunakan pada proses *mining* pada tahap modeling, persiapan data dilakukan dengan sebutan *Preprocessing Data*. *Preprocessing* merupakan hal yang harus dilakukan dalam proses *Data Mining*, karena tidak semua data atau atribut data digunakan dalam proses *Data Mining.* dikarenakan pada penelitian ini digunakan sebuah data *flat File* yang berformat Microsoft Excel agar data tersebut bisa digunakan maka tahapan yang digunakan dalam persiapan yaitu *Ekstraksi data*, *Selecting data* dan *Clean data.*

1. Ekstraksi Data

Ekstraksi data dapat diartikan sebagai proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data ke tingkat selanjutnya ataupun untuk menyimpan data hasil ekstrak tersebut. Dalam penelitian ini, data yang berasal dari *flat File* berformat *microsoft excel* diekstrak, kemudian disimpan ke dalam sebuah *database* agar memudahkan dalam proses pengolahan data. Berikut merupakan contoh data transaksi yang terdapat di Mini Market HD Mart, di mana atribut yang terdapat dalam data tersebut antara lain, Tanggal, Bon, *Barcode*, Nama Barang, HET, Qty.

2. Pemilihan Atribut

Berdasarkan informasi yang ingin didapat oleh pengguna mengenai model barang yang dibeli secara bersamaan, maka atribut yang digunakan dalam data transaksi adalah Bon dan Nama Barang atribut ini akan digunakan untuk memenuhi tujuan awal di mana akan dicari pola pembelian konsumen berdasarkan barang yang di beli. Di bawah ini adalah tabel hasil dari pemilihan atribut yang dapat dilihat pada tabel 3.2 di bawah ini:

Tabel 3.2 Pemilihan atribut

Bon	Nama Barang
HD1000000003506	CHITATO BEB 35G
HD1000000003506	LAY RUMPUT LAUT
HD1000000003506	QTELA KEJU PANGGANG 60G
HD1000000003506	SILVER QUEEN MIDI CASH 33G
HD1000000003509	BLUE BAND SERBAGUNA 200G
HD1000000003510	CHITATO BEB 35G
HD1000000003510	CHITATO ORI FLAVOR 35G
HD1000000003510	CHITATO RASA AYAM BEB 35G
HD1000000003510	HEART
HD1000000003510	SILVER QUEEN MIDI CASH 33G
HD1000000003512	CHITATO RASA AYAM BEB 35G
HD1000000003512	FANTA STROBERI BTL 1L

3. Pembersihan Data

Pembersihan data merupakan proses menghilangkan *noise* berdasarkan dari hasil *task verify data quality*. Data yang dihilangkan adalah data pada satu transaksi yang hanya terdapat satu pembelian. Berikut adalah tabel data yang belum di bersihkan yang dapat dilihat pada tabel 3.3 di bawah:

Tabel 3.3 Tabel yang belum dibersihkan

Bon	Nama Barang
HD1000000003506	CHITATO BEB 35G
HD1000000003506	LAY RUMPUT LAUT
HD1000000003506	QTELA KEJU PANGGANG 60G
HD1000000003506	SILVER QUEEN MIDI CASH 33G
HD1000000003509	BLUE BAND SERBAGUNA 200G

Bon	Nama Barang
HD1000000003510	CHITATO BEB 35G
HD1000000003510	CHITATO ORI FLAVOR 35G
HD1000000003510	CHITATO RASA AYAM BEB 35G
HD1000000003510	HEART
HD1000000003510	SILVER QUEEN MIDI CASH 33G
HD1000000003512	CHITATO RASA AYAM BEB 35G
HD1000000003512	FANTA STROBERI BTL 1L

Untuk tabel yang telah melalui proses pembersihan data dapat dilihat di tabel 3.4 di bawah ini:

Tabel 3.4 Tabel sesudah dibersihkan

Bon	Nama Barang
HD1000000003506	CHITATO BEB 35G
HD1000000003506	LAY RUMPUT LAUT
HD1000000003506	QTELA KEJU PANGGANG 60G
HD1000000003506	SILVER QUEEN MIDI CASH 33G
HD1000000003509	BLUE BAND SERBAGUNA 200G
HD1000000003510	CHITATO BEB 35G
HD1000000003510	CHITATO ORI FLAVOR 35G
HD1000000003510	CHITATO RASA AYAM BEB 35G
HD1000000003510	HEART
HD1000000003510	SILVER QUEEN MIDI CASH 33G
HD1000000003512	CHITATO RASA AYAM BEB 35G
HD1000000003512	FANTA STROBERI BTL 1L

3.1.2.4 Pemodelan

Pemodelan merupakan tahap untuk membuat model atau desain sistem yang akan dibangun.

1. Memilih Teknik Pemodelan

Tujuan dari penelitian ini adalah menemukan pola asosiasi pembelian konsumen, maka dari itu teknik pemodelan yang digunakan adalah Metode *Association Rule* dengan menggunakan Algoritma *CT-Pro*. Dengan menerapkan *Data Mining*, data transaksi akan diolah dengan aturan asosiasi atau metode

Association Rule untuk menemukan pola-pola pembelian barang yang sering dibeli oleh konsumen.

2. Membangun Model

Setelah memilih model yang digunakan tahap selanjutnya adalah membangun model sesuai dengan yang ditentukan. Metodologi dasar *Association Rule* terbagi menjadi dua tahap, yaitu analisa pola frekuensi tinggi dan Pembentukan Aturan Assosiatif. Algoritma yang digunakan dalam penelitian ini adalah Algoritma *CT-Pro*. Berikut ini adalah langkah-langkah proses pengerjaan Algoritma *CT-Pro*:

a. Menghitung Frekuensi Kemunculan Tiap *Item*

Menghitung tiap frekuensi kemunculan tiap *item* merupakan tahapan pertama dalam tahapan metodologi dasar *Association Rule*. Hasil dari perhitungan kemunculan setiap barang dapat dilihat pada tabel 3.5 di bawah:

Tabel 3. 5 *Support Count* Tiap barang

Nama Barang	*Support Count*
COCA COLA BTL IL	16
CHITATO BEB 35G	15
SILVER QUEEN MIDI CASH 33G	15
LAY RUMPUT LAUT	13
POCARI SWEAT BTL 500ML	13
QTELA KEJU PANGGANG 60G	13
FLORIDINA ORG BTL 360ML	11
CHITATO ORI FLAVOR 35G	8
CHITATO RASA AYAM BEB 35G	8
FANTA STROBERI BTL 1L	8
HEART	8
TEBS KALENG 330 ML	8
TROPICANA CHO 70ML	8
LUWAK WHITE KOFFIE 20G	7
BAYGON BAKAR STD	1
LFBUOY TOTAL 10	1
PASEO SMART 250 S	1
SAUS IF SAMBAL PEDAS PET	1
SWEETY FIT PANTZ M8	1

b. Seleksi item dengan Menentukan Nilai *Minimum Support*

Awal mula ditentukan *minimum support* dan juga *confidence* dari keinginan pengguna untuk melihat batasan terendah munculnya barang. Penentuan *minimum support* sangat berpengaruh dengan hasil yang akan didapatkan. Apabila nilai *minimum support* terlalu besar maka hasil yang didapatkan terlalu sedikit bahkan tidak ada, sedangkan apabila nilai *minimum support*nya terlalu kecil, hasil yang didapatkan terlalu banyak. Untuk kasus ini dimisalkan ditentukan *minimum support*: 3% dan *confidence:* 60%. Persentase 3% digunakan sebagai *minimum support* agar terlihat barang yang kemunculannya lebih dari 1 dan dapat dikatakan bahwa menu itu pernah dibeli lebih dari 1x dalam 46 transaksi ini. *Minimum support* dikalikan dengan jumlah transaksi. 0.06 * 46 = 2.76 ≈ 3. Oleh karena itu maka batasan barang yang muncul harus sebanyak >= 3. Setelah itu menu yang *frequent*-nya tidak memenuhi batas *minimum support* dieliminasi lalu diurutkan kembali berdasarkan kemunculan terbanyak dari *frequent*nya.

Langkah selanjutnya adalah menghapus setiap barang yang tidak memenuhi batasan *minimum support* dan diurutkan kembali berdasarkan kemunculan terbanyak dapat dilihat pada tabel 3.6 di bawah ini:

Tabel 3. 6 Hasil Eliminasi *Minimum support*

Nama Barang	*Support Count*
COCA COLA BTL IL	16
CHITATO BEB 35G	15
SILVER QUEEN MIDI CASH 33G	15
LAY RUMPUT LAUT	13
POCARI SWEAT BTL 500ML	13
QTELA KEJU PANGGANG 60G	13
FLORIDINA ORG BTL 360ML	11
CHITATO ORI FLAVOR 35G	8
CHITATO RASA AYAM BEB 35G	8
FANTA STROBERI BTL 1L	8
HEART	8
TEBS KALENG 330 ML	8
TROPICANA CHO 70ML	8
LUWAK WHITE KOFFIE 20G	7

.

Untuk barang yang *frequent*-nya tidak memenuhi batas *minimum support* dieliminasi dan hasilnya dapat dilihat pada tabel 3.7 di bawah ini:

Tabel 3. 7 Data transaksi dengan *item* yang Frekuen

Bon	Nama Barang
HD1000000003506	CHITATO BEB 35G
HD1000000003506	COCA COLA BTL IL
HD1000000003506	FANTA STROBERI BTL 1L
HD1000000003506	LAY RUMPUT LAUT
HD1000000003506	POCARI SWEAT BTL 500ML
HD1000000003506	TROPICANA CHO 70ML
HD1000000003510	CHITATO BEB 35G
HD1000000003510	CHITATO ORI FLAVOR 35G
HD1000000003510	CHITATO RASA AYAM BEB 35G
HD1000000003510	HEART
HD1000000003510	SILVER QUEEN MIDI CASH 33G
HD1000000003512	CHITATO RASA AYAM BEB 35G
HD1000000003512	FANTA STROBERI BTL 1L
HD1000000003512	HEART
HD1000000003512	LAY RUMPUT LAUT
HD1000000003512	LUWAK WHITE KOFFIE 20G
HD1000000003512	TEBS KALENG 330 ML
HD1000000003512	TROPICANA CHO 70ML
HD1000000003513	POCARI SWEAT BTL 500ML
HD1000000003513	QTELA KEJU PANGGANG 60G
HD1000000003514	CHITATO BEB 35G
HD1000000003514	LAY RUMPUT LAUT
HD1000000003514	QTELA KEJU PANGGANG 60G
HD1000000003514	SILVER QUEEN MIDI CASH 33G
HD1000000003515	CHITATO ORI FLAVOR 35G
HD1000000003515	CHITATO RASA AYAM BEB 35G
Bon	Nama Barang
HD1000000003515	QTELA KEJU PANGGANG 60G

Bon	Nama Barang
HD1000000003559	POCARI SWEAT BTL 500ML
HD1000000003564	CHITATO BEB 35G
HD1000000003564	CHITATO ORI FLAVOR 35G
HD1000000003564	FANTA STROBERI BTL 1L
HD1000000003564	POCARI SWEAT BTL 500ML
HD1000000003564	QTELA KEJU PANGGANG 60G
HD1000000003566	CHITATO BEB 35G
HD1000000003566	LAY RUMPUT LAUT
HD1000000003566	POCARI SWEAT BTL 500ML
HD1000000003567	FLORIDINA ORG BTL 360ML
HD1000000003567	LAY RUMPUT LAUT
HD1000000003568	CHITATO BEB 35G
HD1000000003568	COCA COLA BTL IL
HD1000000003568	FANTA STROBERI BTL 1L
HD1000000003568	SILVER QUEEN MIDI CASH 33G
HD1000000003570	FLORIDINA ORG BTL 360ML
HD1000000003570	QTELA KEJU PANGGANG 60G
HD1000000003571	POCARI SWEAT BTL 500ML
HD1000000003571	SILVER QUEEN MIDI CASH 33G
HD1000000003572	CHITATO BEB 35G
HD1000000003572	COCA COLA BTL IL
HD1000000003574	POCARI SWEAT BTL 500ML
HD1000000003574	SILVER QUEEN MIDI CASH 33G
HD1000000003574	QTELA KEJU PANGGANG 60G
HD1000000003582	POCARI SWEAT BTL 500ML
HD1000000003582	SILVER QUEEN MIDI CASH 33G
Bon	Nama Barang
HD1000000003591	COCA COLA BTL IL

Bon	Nama Barang
HD1000000003515	SILVER QUEEN MIDI CASH 33G
HD1000000003515	TEBS KALENG 330 ML
HD1000000003516	CHITATO BEB 35G
HD1000000003516	CHITATO ORI FLAVOR 35G
HD1000000003516	CHITATO RASA AYAM BEB 35G
HD1000000003516	COCA COLA BTL IL
HD1000000003516	TEBS KALENG 330 ML
HD1000000003521	CHITATO BEB 35G
HD1000000003521	POCARI SWEAT BTL 500ML
HD1000000003521	TEBS KALENG 330 ML
HD1000000003521	TROPICANA CHO 70ML
HD1000000003522	FLORIDINA ORG BTL 360ML
HD1000000003522	HEART
HD1000000003522	LAY RUMPUT LAUT
HD1000000003522	SILVER QUEEN MIDI CASH 33G
HD1000000003533	CHITATO BEB 35G
HD1000000003533	COCA COLA BTL IL
HD1000000003533	FANTA STROBERI BTL 1L
HD1000000003533	LAY RUMPUT LAUT
HD1000000003533	TEBS KALENG 330 ML
HD1000000003535	QTELA KEJU PANGGANG 60G
HD1000000003535	SILVER QUEEN MIDI CASH 33G
HD1000000003535	TROPICANA CHO 70ML
HD1000000003536	CHITATO BEB 35G
HD1000000003536	CHITATO ORI FLAVOR 35G
HD1000000003536	CHITATO RASA AYAM BEB 35G
HD1000000003536	COCA COLA BTL IL
HD1000000003536	LAY RUMPUT LAUT
HD1000000003541	CHITATO BEB 35G
HD1000000003541	SILVER QUEEN MIDI CASH 33G
HD1000000003542	CHITATO RASA AYAM BEB 35G
HD1000000003542	COCA COLA BTL IL
HD1000000003542	FLORIDINA ORG BTL 360ML

Bon	Nama Barang
HD1000000003591	CHITATO BEB 35G
HD1000000003591	POCARI SWEAT BTL 500ML
HD1000000003597	CHITATO ORI FLAVOR 35G
HD1000000003597	CHITATO ORI FLAVOR 35G
HD1000000003598	COCA COLA BTL IL
HD1000000003598	LUWAK WHITE KOFFIE 20G
HD1000000003598	TEBS KALENG 330 ML
HD1000000003605	CHITATO RASA AYAM BEB 35G
HD1000000003605	COCA COLA BTL IL
HD1000000003605	FLORIDINA ORG BTL 360ML
HD1000000003605	TROPICANA CHO 70ML
HD1000000003610	COCA COLA BTL IL
HD1000000003610	FANTA STROBERI BTL 1L
HD1000000003610	QTELA KEJU PANGGANG 60G
HD1000000003611	QTELA KEJU PANGGANG 60G
HD1000000003611	SILVER QUEEN MIDI CASH 33G
HD1000000003612	FLORIDINA ORG BTL 360ML
HD1000000003612	HEART
HD1000000003612	LAY RUMPUT LAUT
HD1000000003612	TROPICANA CHO 70ML
HD1000000003615	COCA COLA BTL IL
HD1000000003615	HEART
HD1000000003643	FANTA STROBERI BTL 1L
HD1000000003643	HEART
HD1000000003643	TROPICANA CHO 70ML
HD1000000003652	COCA COLA BTL IL
HD1000000003652	LUWAK WHITE KOFFIE 20G
HD1000000003652	QTELA KEJU PANGGANG 60G
HD1000000003654	COCA COLA BTL IL
HD1000000003654	FANTA STROBERI BTL 1L
HD1000000003654	FLORIDINA ORG BTL 360ML
HD1000000003654	LUWAK WHITE KOFFIE 20G
HD1000000003656	COCA COLA BTL IL

Bon	Nama Barang
HD1000000003545	LAY RUMPUT LAUT
HD1000000003545	SILVER QUEEN MIDI CASH 33G
HD1000000003546	CHITATO BEB 35G
HD1000000003546	FLORIDINA ORG BTL 360ML
HD1000000003546	POCARI SWEAT BTL 500ML
HD1000000003546	SILVER QUEEN MIDI CASH 33G
HD1000000003546	TEBS KALENG 330 ML
HD1000000003549	POCARI SWEAT BTL 500ML
HD1000000003549	QTELA KEJU PANGGANG 60G
HD1000000003549	SILVER QUEEN MIDI CASH 33G
HD1000000003553	CHITATO BEB 35G
HD1000000003553	POCARI SWEAT BTL 500ML
HD1000000003554	FLORIDINA ORG BTL 360ML
HD1000000003554	LAY RUMPUT LAUT
HD1000000003559	SILVER QUEEN MIDI CASH 33G
HD1000000003559	FLORIDINA ORG BTL 360ML

Bon	Nama Barang
HD1000000003656	LAY RUMPUT LAUT
HD1000000003656	LUWAK WHITE KOFFIE 20G
HD1000000003656	QTELA KEJU PANGGANG 60G
HD1000000003656	TROPICANA CHO 70ML
HD1000000003660	FLORIDINA ORG BTL 360ML
HD1000000003660	HEART
HD1000000003660	LAY RUMPUT LAUT
HD1000000003660	LUWAK WHITE KOFFIE 20G
HD1000000003660	QTELA KEJU PANGGANG 60G
HD1000000003662	CHITATO ORI FLAVOR 35G
HD1000000003662	CHITATO RASA AYAM BEB 35G
HD1000000003662	COCA COLA BTL IL
HD1000000003662	HEART
HD1000000003662	LUWAK WHITE KOFFIE 20G
HD1000000003662	TEBS KALENG 330 ML

c. Membangun *Global Item tabel*

Setelah mengeliminasi *item* yang tidak memenuhi *minimum support* selanjutnya adalah memberikan *Id Global item* pada *item* yang memenuhi *minimum support* dengan memberikan penomoran secara *ascending* dari frekuensi terbesar ke terkecil. Berikut adalah tabel *Global item* yang dapat dilihat pada tabel. Berikut adalah tabel *Global item* yang dapat dilihat pada tabel 3.8 di bawah:

Tabel 3. 8 *Global Item tabel*

Nama barang	*Support count*	*Id Globalitem*
COCA COLA BTL IL	16	1
CHITATO BEB 35G	15	2
SILVER QUEEN MIDI CASH 33G	15	3
QTELA KEJU PANGGANG 60G	13	4
POCARI SWEAT BTL 500ML	13	5
LAY RUMPUT LAUT	13	6

Nama barang	*Support count*	*Id Globalitem*
FLORIDINA ORG BTL 360ML	11	7
FANTA STROBERI BTL 1L	8	8
TROPICANA CHO 70ML	8	9
TEBS KALENG 330 ML	8	10
CHITATO ORI FLAVOR 35G	8	11
CHITATO RASA AYAM BEB 35G	8	12
HEART	8	13
LUWAK WHITE KOFFIE 20G	7	14

d. *Mapping data*

Setelah Membangun *Global Item tabel* selanjutnya yaitu *Mapping* data. *Mapping* yaitu memetakan data transaksi dari tabel 3.9 terhadap *id Global item* pada tabel 3.8, di mana nama barang pada tabel 3.8 digantikan oleh *id Global item* sesuai dengan yang ada pada tabel 3.8. Data transaksi kemudian diurutkan dari *id Global item* terkecil ke terbesar dari setiap Bon. Untuk hasil pemberian *id GlobalItemTable* dan pengurutan dapat dilihat pada tabel 3.11.

Tabel 3. 9 Tabel hasil *Mapping*

Bon	Nama Barang
HD1000000003506	1
HD1000000003506	2
HD1000000003506	5
HD1000000003506	6
HD1000000003506	8
HD1000000003506	9
HD1000000003510	2
HD1000000003510	3
HD1000000003510	11
HD1000000003510	12
HD1000000003510	13
HD1000000003512	6
HD1000000003512	8
HD1000000003512	9
HD1000000003512	10
HD1000000003512	12
HD1000000003512	13

Bon	Nama Barang
HD1000000003559	7
HD1000000003564	2
HD1000000003564	4
HD1000000003564	5
HD1000000003564	8
HD1000000003564	11
HD1000000003566	2
HD1000000003566	5
HD1000000003566	6
HD1000000003567	6
HD1000000003567	7
HD1000000003568	1
HD1000000003568	2
HD1000000003568	3
HD1000000003568	8
HD1000000003570	4
HD1000000003570	7

Bon	Nama Barang
HD1000000003512	14
HD1000000003513	2
HD1000000003513	3
HD1000000003514	4
HD1000000003514	6
HD1000000003514	4
HD1000000003514	5
HD1000000003515	3
HD1000000003515	4
HD1000000003515	10
HD1000000003515	11
HD1000000003515	12
HD1000000003516	1
HD1000000003516	2
HD1000000003516	11
HD1000000003516	10
HD1000000003516	12
HD1000000003521	2
HD1000000003521	5
HD1000000003521	9
HD1000000003521	10
HD1000000003522	3
HD1000000003522	6
HD1000000003522	7
HD1000000003522	13
HD1000000003533	1
HD1000000003533	2
HD1000000003533	8
HD1000000003533	6
HD1000000003533	10
HD1000000003535	3
HD1000000003535	4
HD1000000003535	9
HD1000000003536	1
HD1000000003536	2
HD1000000003536	6
HD1000000003536	11
HD1000000003536	12
HD1000000003541	2
HD1000000003541	3

Bon	Nama Barang
HD1000000003571	3
HD1000000003571	5
HD1000000003572	1
HD1000000003572	2
HD1000000003574	3
HD1000000003574	4
HD1000000003574	5
HD1000000003582	3
HD1000000003582	5
HD1000000003591	1
HD1000000003591	2
HD1000000003591	5
HD1000000003597	11
HD1000000003597	1
HD1000000003598	10
HD1000000003598	11
HD1000000003598	14
HD1000000003605	1
HD1000000003605	7
HD1000000003605	9
HD1000000003605	12
HD1000000003610	1
HD1000000003610	4
HD1000000003610	8
HD1000000003611	3
HD1000000003611	4
HD1000000003612	6
HD1000000003612	7
HD1000000003612	9
HD1000000003612	13
HD1000000003615	1
HD1000000003615	13
HD1000000003643	8
HD1000000003643	9
HD1000000003643	13
HD1000000003652	1
HD1000000003652	4
HD1000000003652	14
HD1000000003654	1
HD1000000003654	7

Bon	Nama Barang
HD1000000003542	1
HD1000000003542	7
HD1000000003542	12
HD1000000003545	3
HD1000000003545	6
HD1000000003546	2
HD1000000003546	3
HD1000000003546	5
HD1000000003546	7
HD1000000003546	10
HD1000000003549	3
HD1000000003549	4
HD1000000003549	5
HD1000000003553	2
HD1000000003553	5
HD1000000003554	6
HD1000000003554	7
HD1000000003559	3
HD1000000003559	5

Bon	Nama Barang
HD1000000003654	8
HD1000000003654	14
HD1000000003656	1
HD1000000003656	4
HD1000000003656	6
HD1000000003656	9
HD1000000003656	14
HD1000000003660	4
HD1000000003660	6
HD1000000003660	7
HD1000000003660	13
HD1000000003660	14
HD1000000003662	1
HD1000000003662	10
HD1000000003662	11
HD1000000003662	12
HD1000000003662	13
HD1000000003662	14

e. Pembentukan CFP – *Tree*

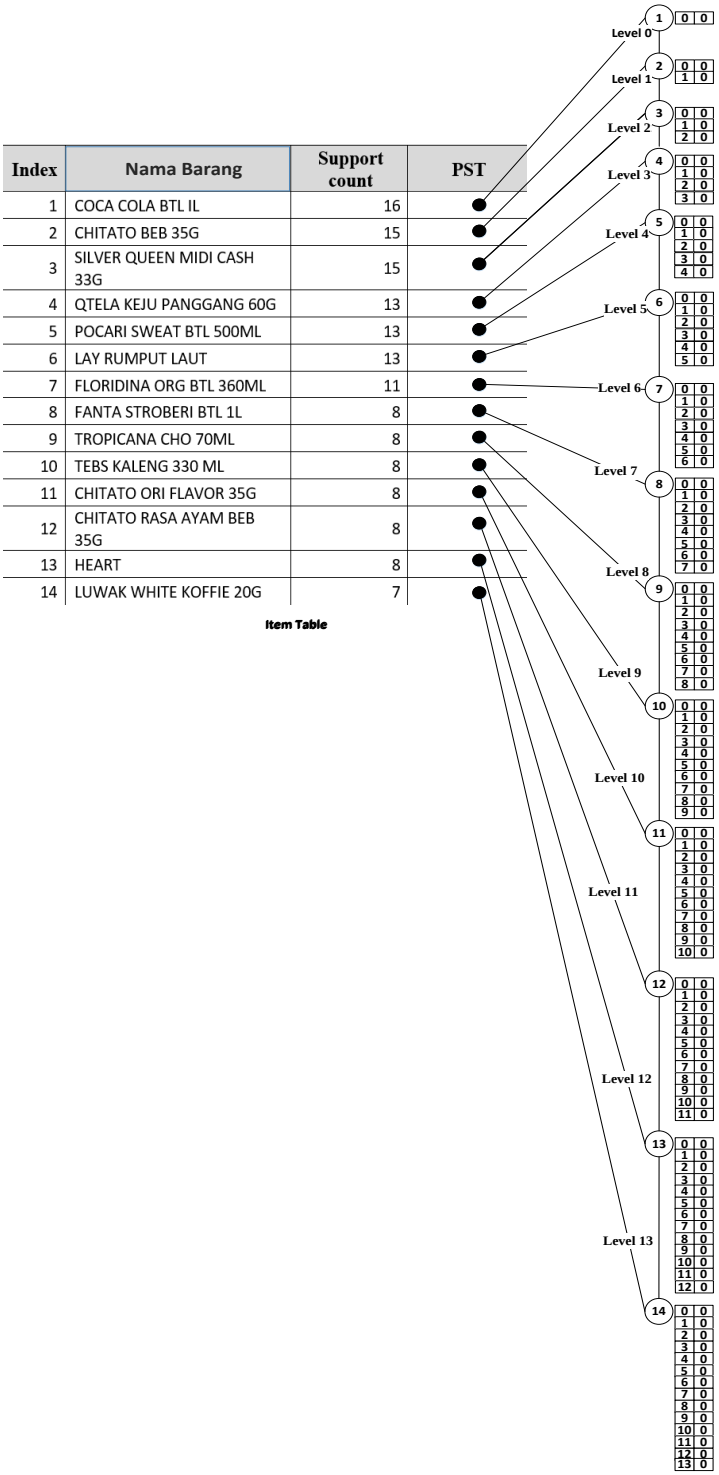
Setelah proses *Mapping* selesai kemudian ke tahap selanjutnya adalah *inset to CFP-Tree* yaitu memasukkan setiap data transaksi ke dalam *tree,* berikut merupakan tahapannya:

1. Tahapan pertama dalam pembentukan *CFP-Tree* adalah dengan membuat *left branch* yaitu dengan men-*generate node* mulai dari *index* pertama sampai ke n berdasarkan *index* yang pada *headertable,* selanjutnya hasil dari *generate node* akan dihubungkan dengan *headertable* menggunakan *pointer*
2. Dimulai dari item yang pertama kali muncul dari no. transaksi pertama yang telah di *Mapping,* item tersebut akan menjadi *currnode.*
3. Tambah *count* = 1 pada item pertama muncul sesuai dengan levelnya.
4. Jika *currnode* sudah memiliki anak, dan item yang muncul sama dengan anak tersebut maka tambahkan *count* =1 pada level yang sama *corrnode.*
5. Jika *currnode* sudah memiliki anak, tetapi data yang muncul berbeda dengan anak dari *currnode,* buat *node* baru dan tambah *count* = 1 pada level yang sama dengan *currnode*, hubungkan *node* baru tersebut pada *currnode* melalui *nodelink.*

Berikut merupakan hasil pembentukan *leftmost branch* dapat dilihat pada gambar 3.3

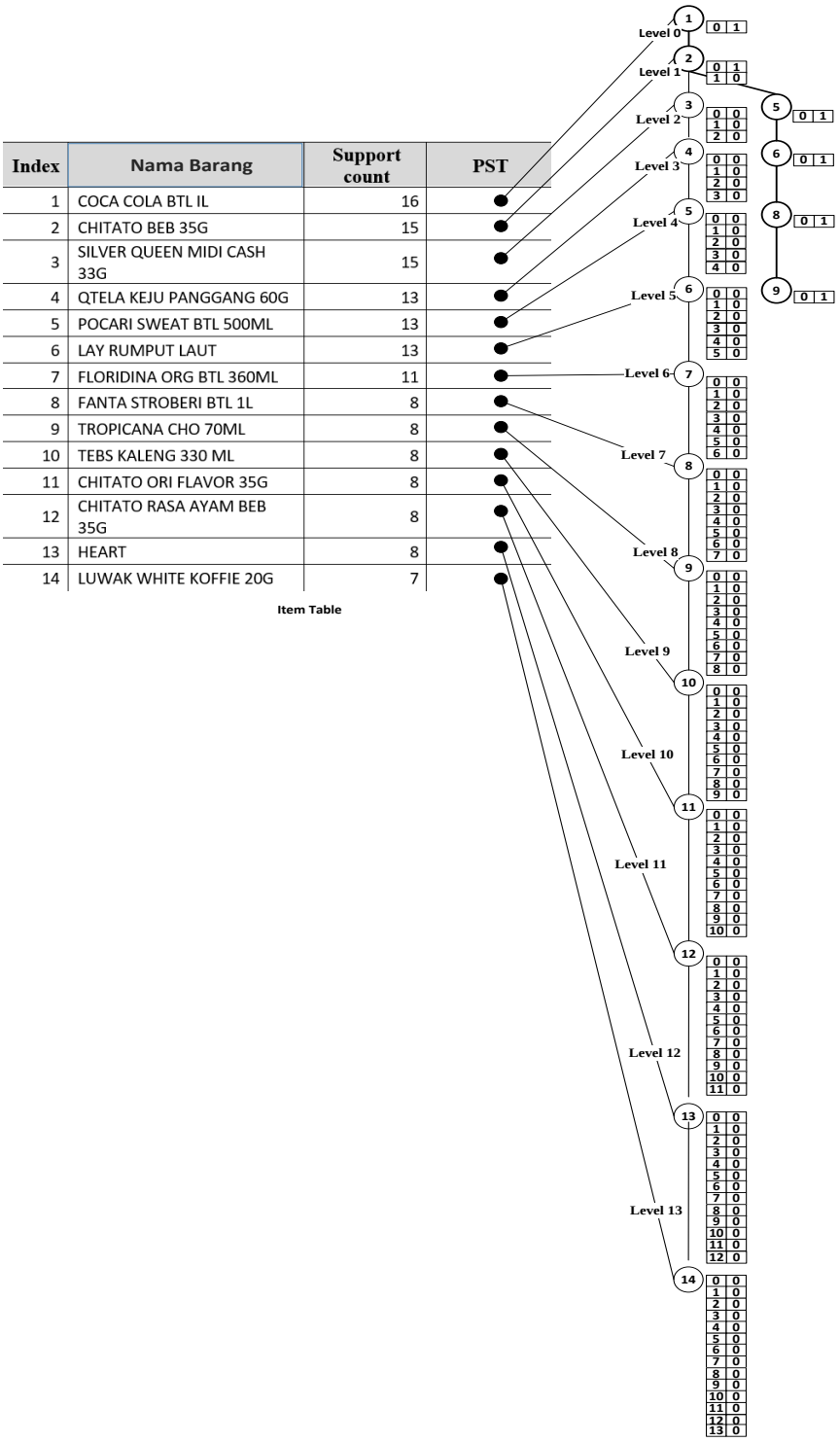
Index	Nama Barang	Support count	PST
1	COCA COLA BTL IL	16	
2	CHITATO BEB 35G	15	
3	SILVER QUEEN MIDI CASH 33G	15	
4	QTELA KEJU PANGGANG 60G	13	
5	POCARI SWEAT BTL 500ML	13	
6	LAY RUMPUT LAUT	13	
7	FLORIDINA ORG BTL 360ML	11	
8	FANTA STROBERI BTL 1L	8	
9	TROPICANA CHO 70ML	8	
10	TEBS KALENG 330 ML	8	
11	CHITATO ORI FLAVOR 35G	8	
12	CHITATO RASA AYAM BEB 35G	8	
13	HEART	8	
14	LUWAK WHITE KOFFIE 20G	7	

Item Table



Gambar 3. 3 Hasil pembentukan *left most branch tree*

1. Bon HD1000000003506 dengan *id global* 1,2,5,6,8 dan 9 dapat dilihat pada gambar 3.4 di bawah ini:

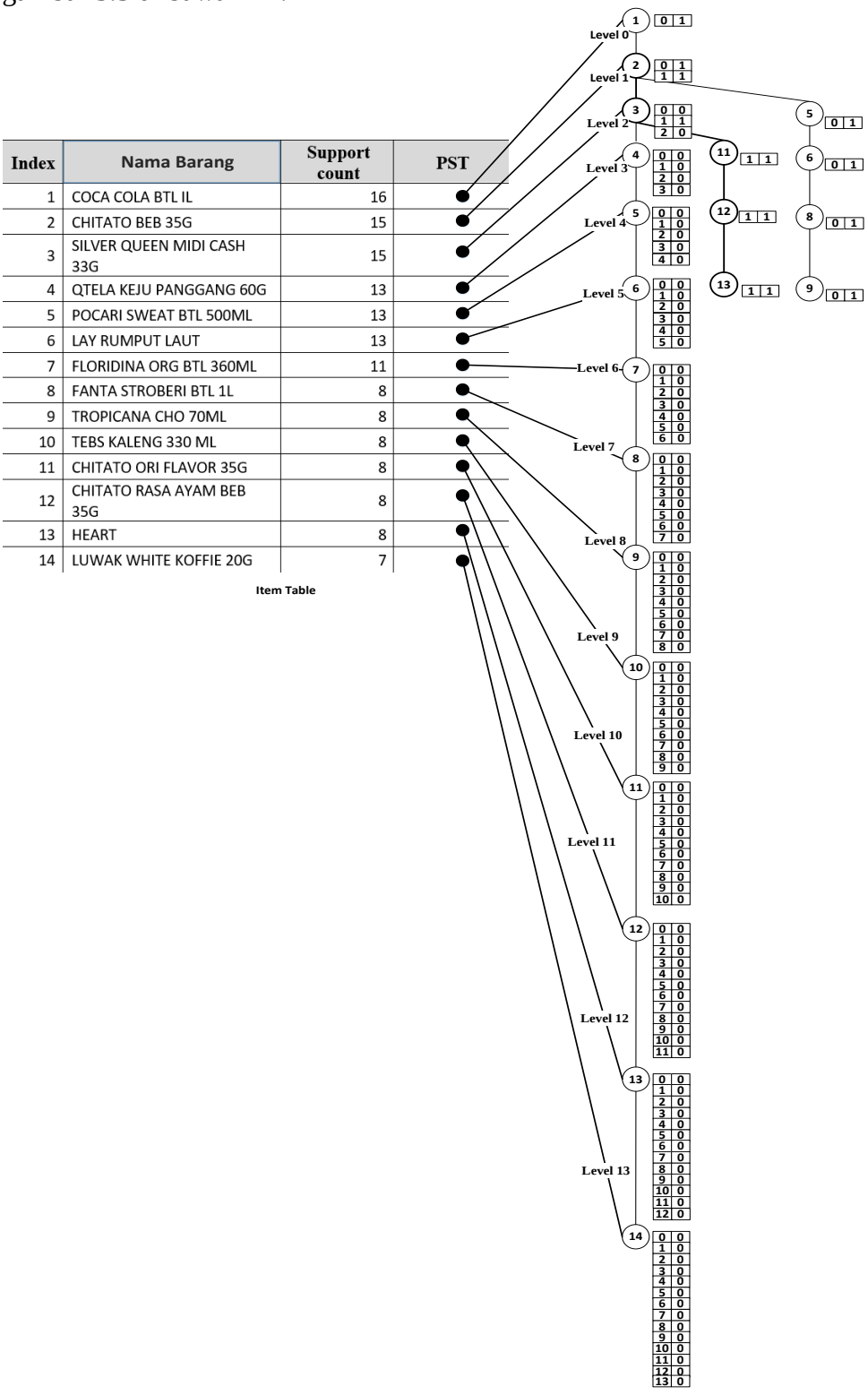


Index	Nama Barang	Support count	PST
1	COCA COLA BTL IL	16	
2	CHITATO BEB 35G	15	
3	SILVER QUEEN MIDI CASH 33G	15	
4	QTELA KEJU PANGGANG 60G	13	
5	POCARI SWEAT BTL 500ML	13	
6	LAY RUMPUT LAUT	13	
7	FLORIDINA ORG BTL 360ML	11	
8	FANTA STROBERI BTL 1L	8	
9	TROPICANA CHO 70ML	8	
10	TEBS KALENG 330 ML	8	
11	CHITATO ORI FLAVOR 35G	8	
12	CHITATO RASA AYAM BEB 35G	8	
13	HEART	8	
14	LUWAK WHITE KOFFIE 20G	7	

Item Table

Gambar 3. 4 Pembentukan *CFP-Tree* Bon HD1000000003506

2. Bon HD1000000003506 dengan *id global* 2,3,11,12 dan 13 dapat dilihat pada gambar 3.5 di bawah ini:

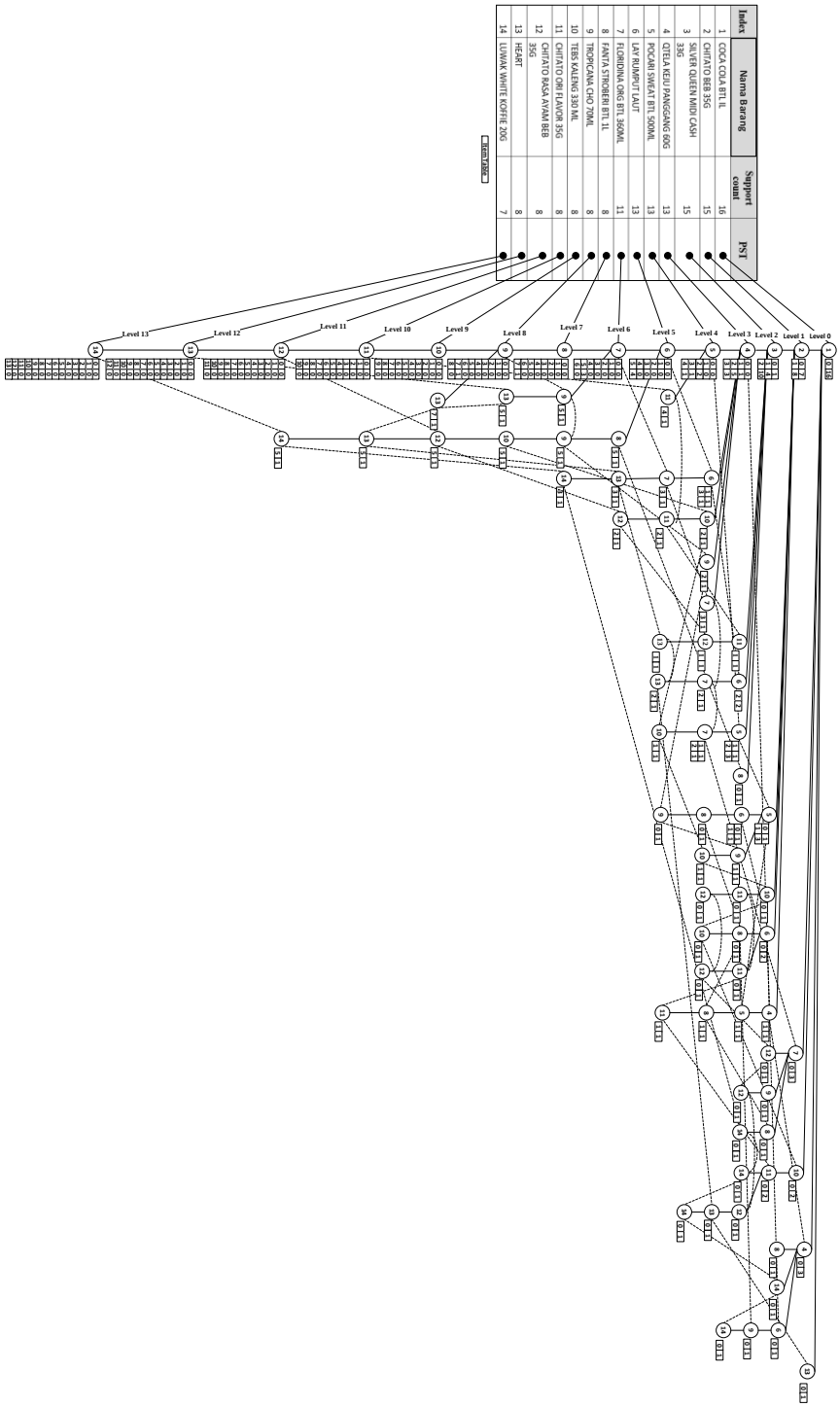


Index	Nama Barang	Support count	PST
1	COCA COLA BTL IL	16	
2	CHITATO BEB 35G	15	
3	SILVER QUEEN MIDI CASH 33G	15	
4	QTELA KEJU PANGGANG 60G	13	
5	POCARI SWEAT BTL 500ML	13	
6	LAY RUMPUT LAUT	13	
7	FLORIDINA ORG BTL 360ML	11	
8	FANTA STROBERI BTL 1L	8	
9	TROPICANA CHO 70ML	8	
10	TEBS KALENG 330 ML	8	
11	CHITATO ORI FLAVOR 35G	8	
12	CHITATO RASA AYAM BEB 35G	8	
13	HEART	8	
14	LUWAK WHITE KOFFIE 20G	7	

Item Table

Gambar 3. 5 Pembentukan *CFP-Tree* Bon HD1000000003510

3. Setelah pembacaan sampai ke Bon HD1000000003662 dengan *id global* 1,10,11,12,13 dan 14 akan membentuk *Global CFP-Tree* yang dapat dilihat pada gambar 3.6 di bawah:



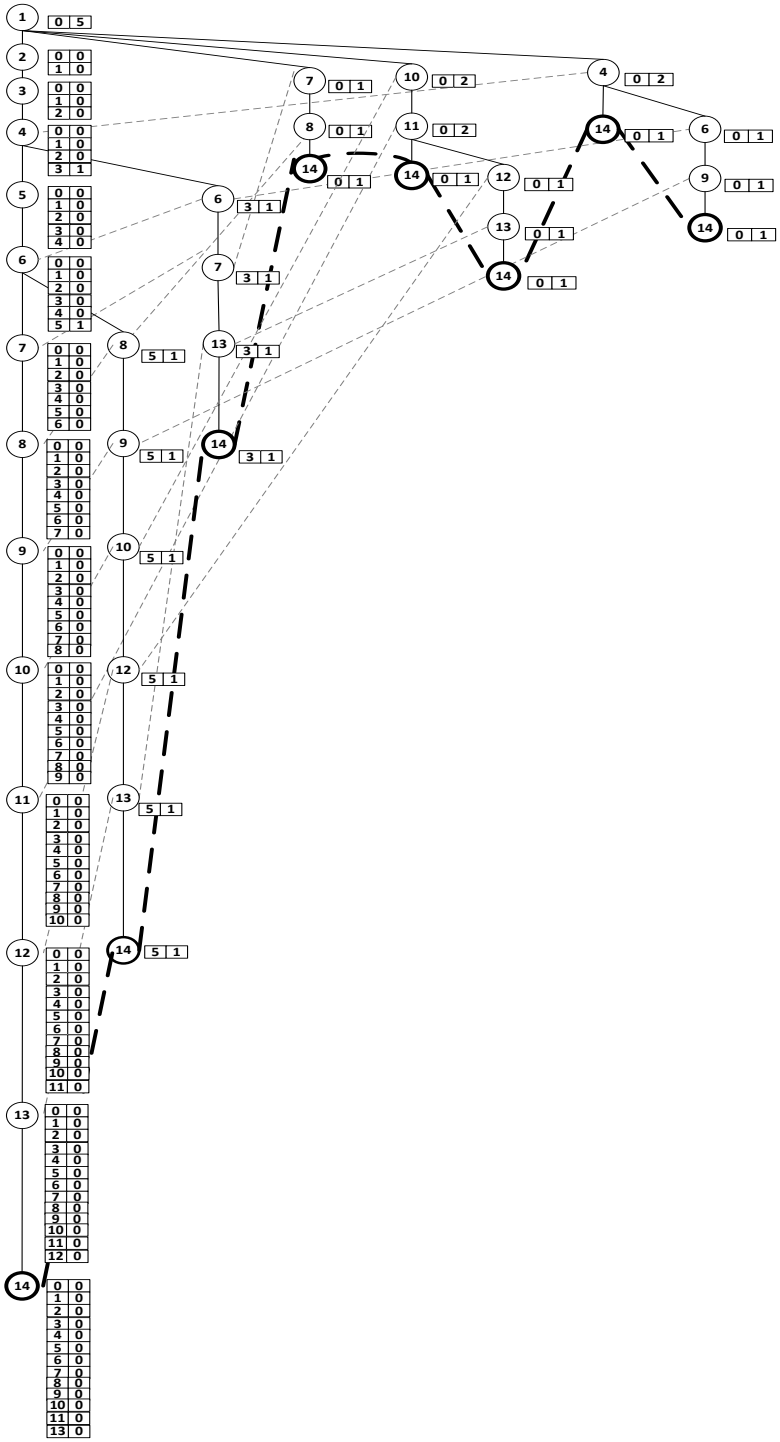
Gambar 3. 6 *Global CFP-Tree*

f. Penggalian *Frequent Patterns*

Setelah tahap pembangunan *CFP-Tree* dari sekumpulan data transaksi, tahap selanjutnya adalah membangun *local* CFP - *Tree* untuk mencari *frequent itemset* yang lebih signifikan. Berikut adalah langkah-langkah dari algoritma CT-Pro:

1. Dimulai dengan *index* dengan *support* count terkecil (x) dari *Globa item Table.*
2. Lakukan penelusuran *node* yang berkaitan dengan *index* x dengan menghitung *count* = 1 pada setiap *node* yang dikunjungi mulai dari *Path* sampai ke *root* untuk mencari *node* yang memenuhi *minimum support count*.
3. Masukan setiap *item* yang memenuhi *minimum support* ke *localitemTable.*
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 di mana x akan menjadi *root* dan *index* yang ada di *local localCFP-Tree* menjadi anaknya.
8. Lakukan penelusuran di *Local CFP-Tree* ulang 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 Pattern Tree index* 14. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 14 secara *bottom-up* yaitu penelusuran mulai dari *Path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.7 di bawah ini:



Gambar 3. 7 *CFP-Tree* berakhiran *index* 14

Berikut hasil penelusuran untuk *index* 14 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.10 di bawah:

Tabel 3. 10 Hasil penelusuran *index* 14

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(5), 4(3), 6(3), 7(2), 8(2), 9(2), 10(3), 11(2), 12(2), 13(3), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(5), 4(3), 6(3), 10(3), 13(3). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table* dan diberikan *index* baru di *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 3.11 di bawah:

Tabel 3.11 *Local item tabel*

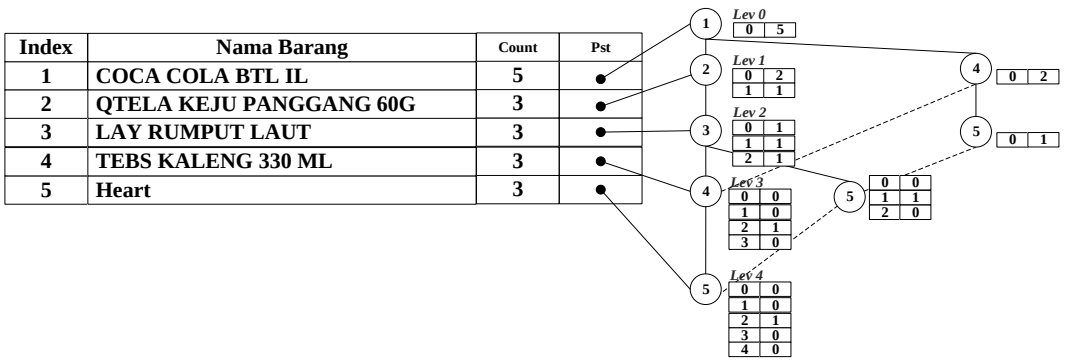
Index local	*Index global*	Nama Barang	*Count*
1	1	COCA COLA BTL IL	5
2	4	QTELA KEJU PANGGANG 60G	3
3	6	LAY RUMPUT LAUT	3
4	10	TEBS KALENG 330 ML	3
5	13	Heart	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.12 di bawah:

Tabel 3. 12 Hasil *Mapping index local*

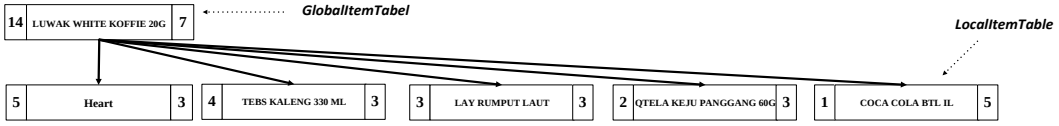
No.	*Path*
1	3,4,5
2	2,3,5
3	1
4	1,4
5	1,4,5
6	1,2
7	1,2,3

Setelah proses *Mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. 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 Table* yang terbentuk dari *Global Item Table* data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local Item Table* yang terbentuk dari *Local Frequent item*. Berikut adalah *Local CFP-Tree* dari *index* 14 dapat dilihat pada gambar 3.8 di bawah:



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

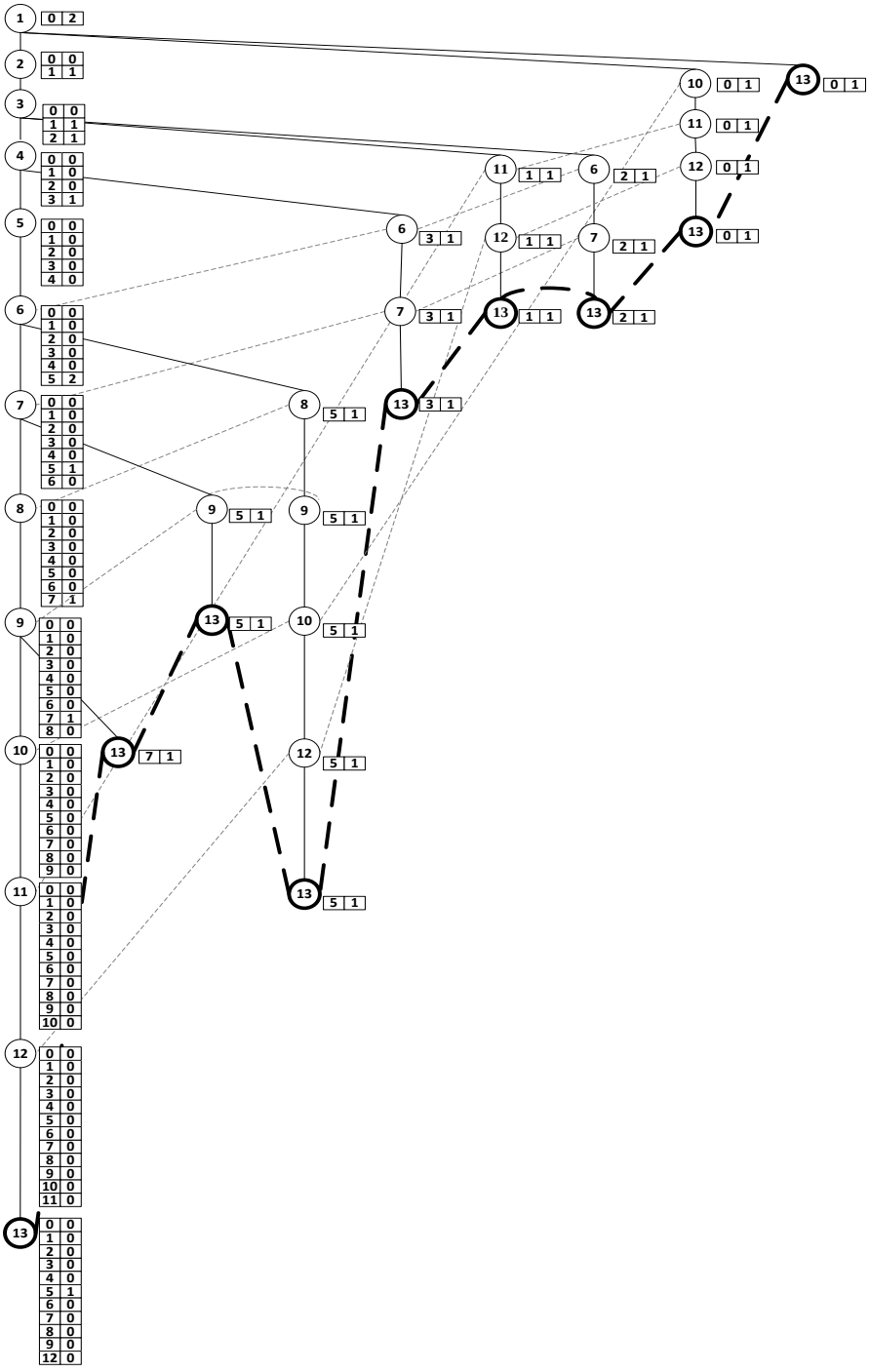
Setelah *Local CFP-Tree index* 14 terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index* 14 akan menjadi *root* dan setiap *index* di *Local item tabel* akan menjadi anak dari *index* 14. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 14 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.9 di bawah:



Gambar 3. 9 *Local CFP-Tree projection* 14

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 14 dengan nama Luwak White Coffie 20G adalah sebagai berikut: (14,5:3), (14,4:3), (14,3:3), (14,1:4).

- Untuk pencarian *local Frequent Pattern Tree index* 13. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 13 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.10 di bawah:



Gambar 3. 10 *CFP-Tree* berakhiran *index* 13

Berikut hasil penelusuran untuk *index* 13 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.13 di bawah:

Tabel 3. 13 Hasil penelusuran *index* 13

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(2), 2(1), 3(2), 4(2), 6(4), 7(3), 8(1), 9(3), 10(2), 11(1), 12(3), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 6(4), 7(3), 9(3), 12(3). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table* dan diberikan *index* baru di *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk pada tabel 3.14 di bawah:

Tabel 3. 14 *Local item tabel*

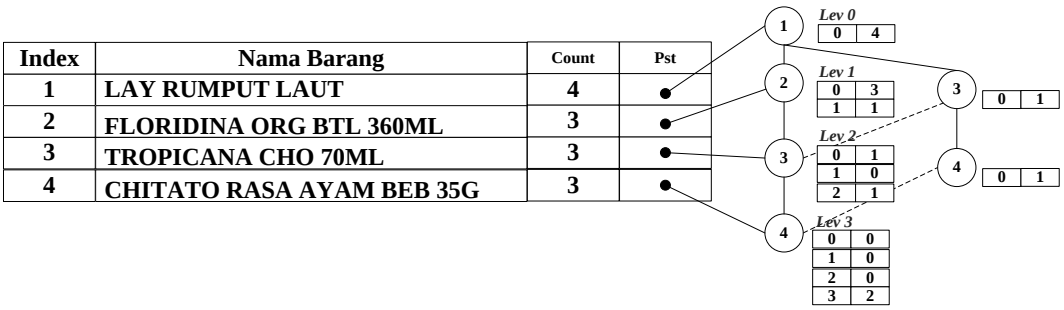
Index local	*Index global*	Nama Barang	*Count*
1	6	LAY RUMPUT LAUT	4
2	7	FLORIDINA ORG BTL 360ML	3
3	9	TROPICANA CHO 70ML	3
4	12	CHITATO RASA AYAM BEB 35G	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.15 di bawah:

Tabel 3. 15 Hasil *Mapping index local*

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

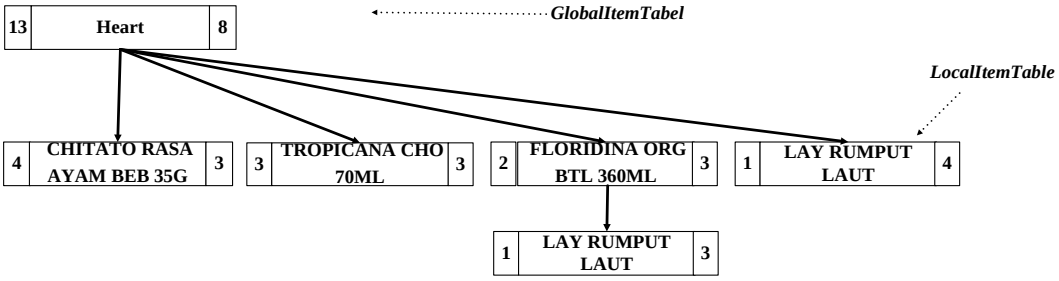
Setelah dilakukan maping data dibuat *Local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item tabel* yang terbentuk dari *Global Item tabel* data sedangkan 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 index* 13 dapat dilihat pada gambar 3.11 di bawah:



Gambar 3. 11 *Local CFP-Tree index* 13

Setelah *Local CFP-Tree index* 13 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 13 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 13. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree*

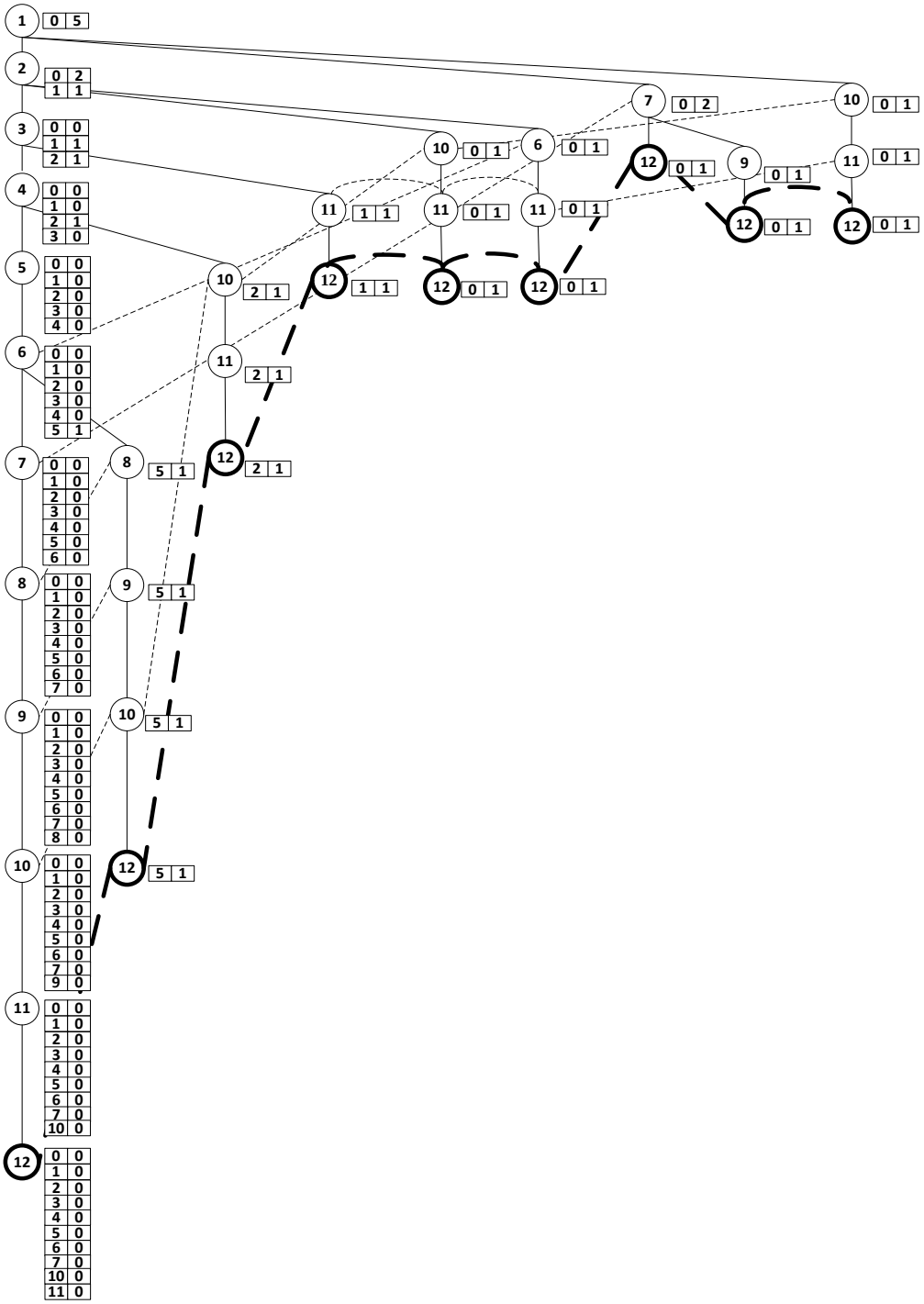
index 13 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.12 di bawah:



Gambar 3. 12 *Local CFP-Tree projection* 13

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 13 dengan nama HEART adalah sebagai berikut: (13,4:3), (13,3:3), (13,2:3), (13,2,1:3),(13,1:4).

- Untuk pencarian *local Frequent Pattern Tree index* 12. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 12 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.13 di bawah:



Gambar 3. 13 *CFP-Tree* berakhiran *index* 12

Berikut hasil penelusuran untuk *index* 12 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.16 di bawah:

Tabel 3. 16 Hasil penelusuran *index* 12

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(5),2(3),3(2),4(1),6(2),7(2),8(1),9(2),10(4),11(5), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(5), 2(3), 10(4), 11(5). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table* dan diberikan *index* baru di *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 3.17 di bawah:

Tabel 3. 17 *Local item tabel*

Index local	*Index global*	Nama Barang	*Count*
1	1	COCA COLA BTL IL	5
2	11	CHITATO ORI FLAVOR 35G	5
3	10	TEBS KALENG 330 ML	4
4	2	CHITATO BEB 35G	3

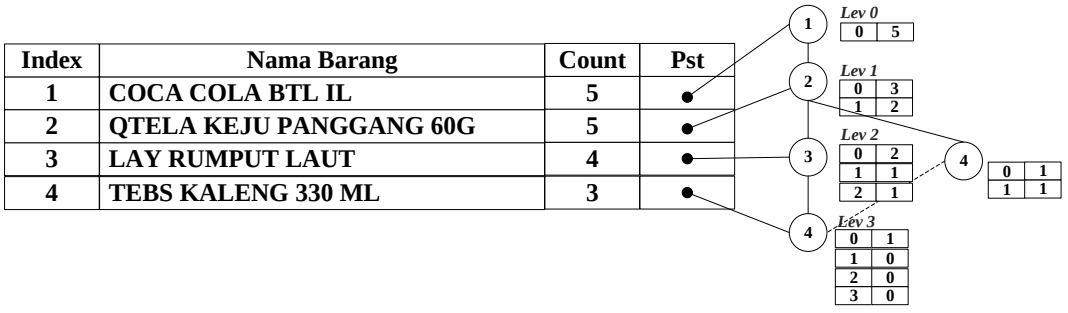
Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.18:

Tabel 3. 18 Hasil *Mapping index local*

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

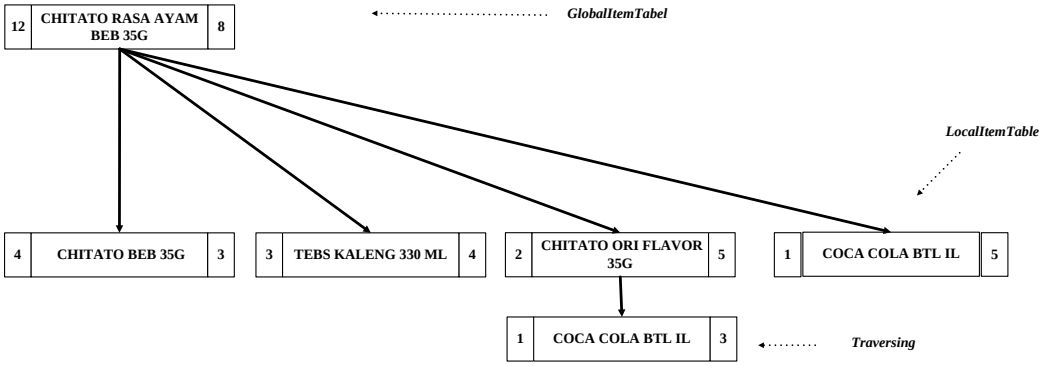
Setelah dilakukan maping data dibuat *Local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item tabel* yang terbentuk dari *Global Item tabel* data sedangkan 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 index* 12 dapat dilihat pada gambar 3.14 di bawah:

Index	Nama Barang	Count	Pst
1	COCA COLA BTL IL	5	
2	QTELA KEJU PANGGANG 60G	5	
3	LAY RUMPUT LAUT	4	
4	TEBS KALENG 330 ML	3	



Gambar 3. 14 *Local CFP-Tree index* 12

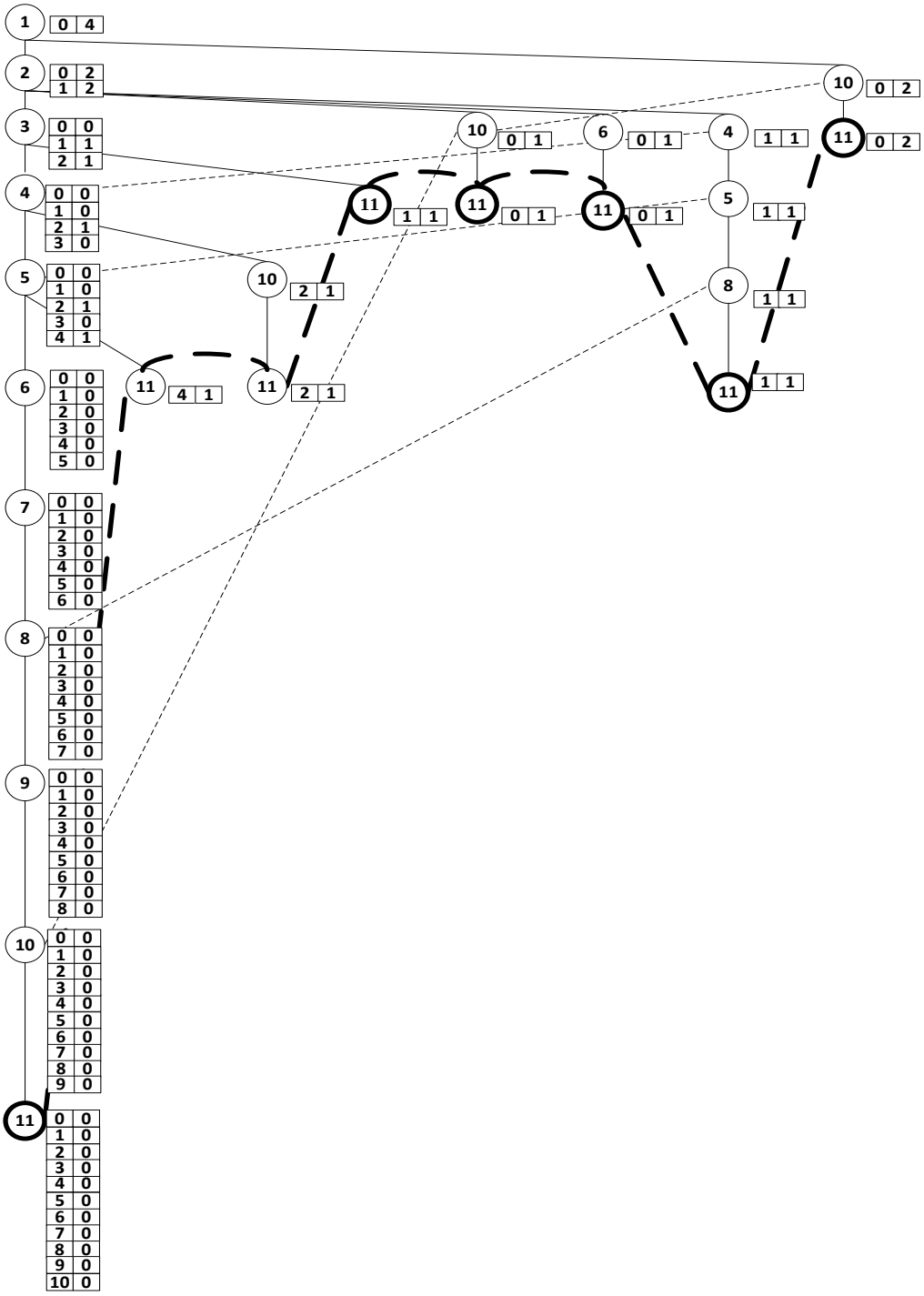
Setelah *Local CFP-Tree index* 12 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 12 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 12. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 12 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.15 di bawah:



Gambar 3. 15 *Local CFP-Tree projection* 12

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 12 dengan nama CHITATO RASA AYAM BEB 35G adalah sebagai berikut: (12,4:3), (12,3:4), (12,2:5), (12,2,1:3), (12,1:5)

- Untuk pencarian *local Frequent Pattern Tree index* 11. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 11 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.16 di bawah:



Gambar 3. 16 *CFP-Tree* berakhiran *index* 11

Berikut hasil penelusuran untuk *index* 11 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.19 di bawah:

Tabel 3. 19 Hasil penelusuran *index* 11

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(4),2(4),4(2),5(1),6(1),8(1),10(4), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(4),2(4),10(4). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.20 di bawah:

Tabel 3. 20 *Local item tabel*

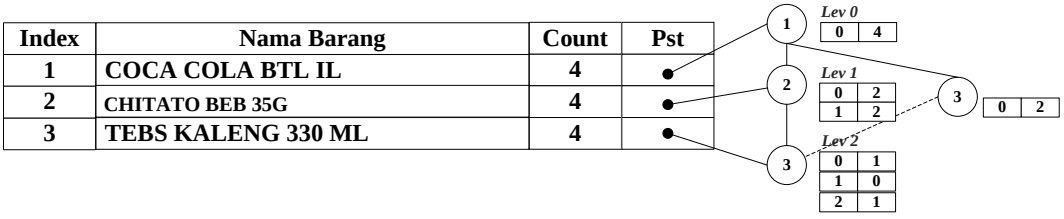
Index local	*Index global*	Nama Barang	*Count*
1	1	COCA COLA BTL IL	4
2	2	CHITATO BEB 35G	4
3	10	TEBS KALENG 330 ML	4

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.21 berikut:

Tabel 3. 21 Hasil *Mapping index local*

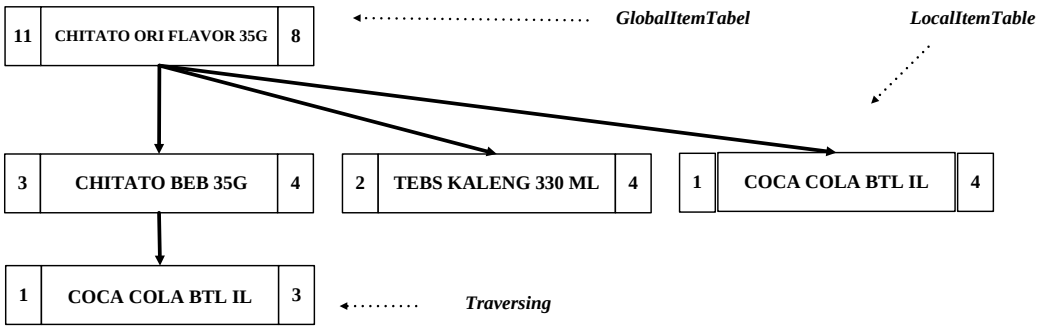
No.	*Path*
1	3
2	2
3	1,2,3
4	1,2
5	2
6	1,3
7	1,3

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 11 dapat dilihat pada gambar 3.17 di bawah:



Gambar 3. 17 *Local CFP-Tree index* 11

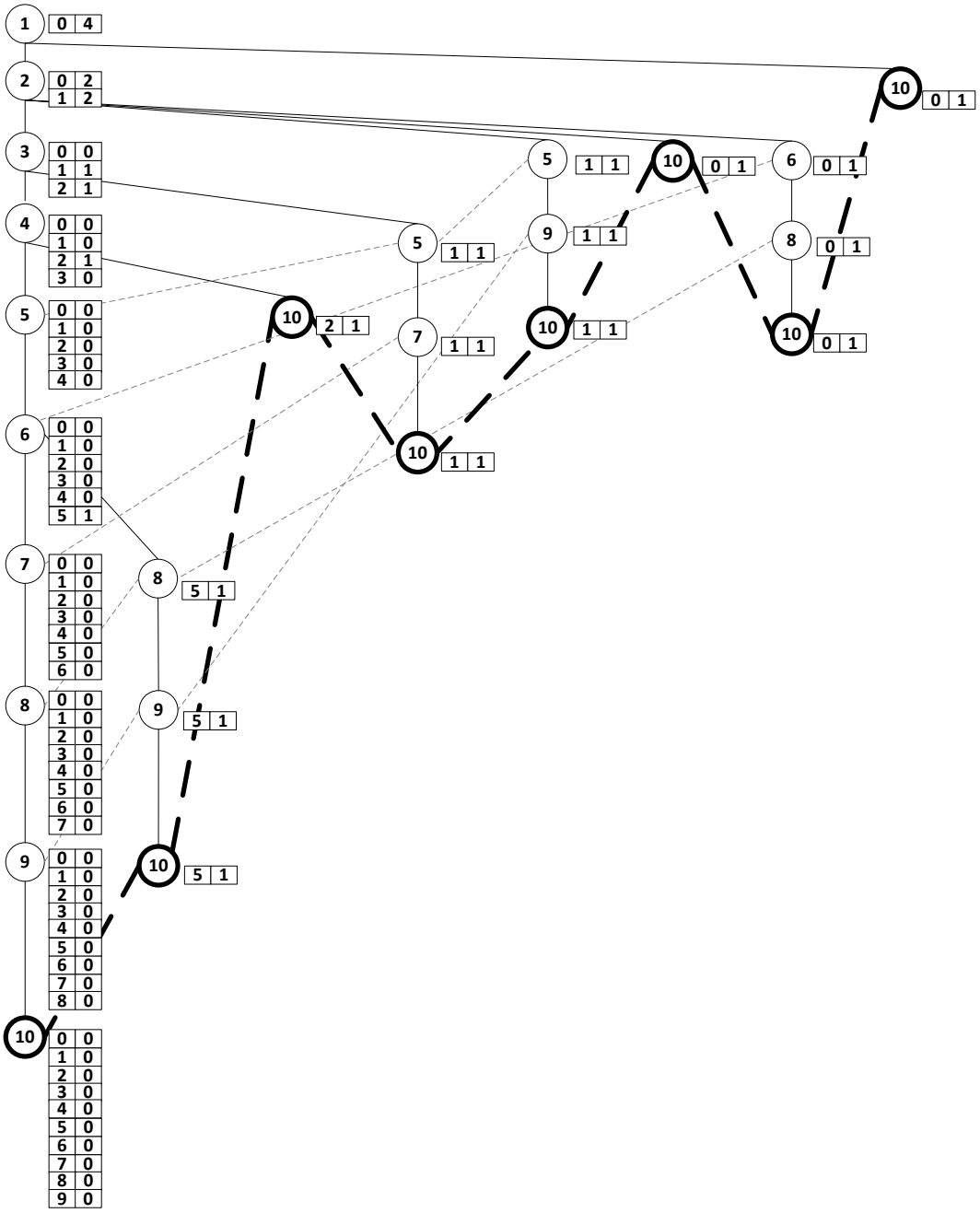
Setelah *Local CFP-Tree index* 11 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 11 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 11. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 11 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.18 berikut:



Gambar 3. 18 *Local CFP-Tree projection* 11

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 11 dengan nama CHITATO ORI FLAVOR 35G adalah sebagai berikut: (11,3:4), (11,3,1:4), (11,2:4), (11,1:4).

- Untuk pencarian *local Frequent Pattern Tree index* 10. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 10 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.19 di bawah:



Gambar 3. 19 *CFP-Tree* berakhiran *index* 10

Berikut hasil penelusuran untuk *index* 10 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.22 berikut:

Tabel 3. 22 Hasil penelusuran *index* 10

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(4),2(4),3(2),4(1),5(2),6(2),7(1),8(2),9(2), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(4),2(4). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.23 di bawah:

Tabel 3. 23 *Local item tabel*

Index local	*Index global*	Nama Barang	*Count*
1	1	CHITATO BEB 35G	4
2	2	COCA COLA BTL IL	4

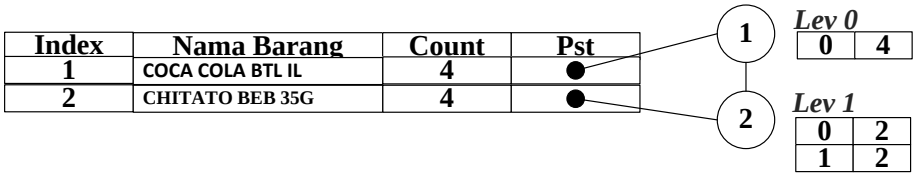
Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.24 di bawah:

Tabel 3. 24 Hasil *Mapping index local*

No.	*Path*
1	2
2	2
3	1,2
4	1,2

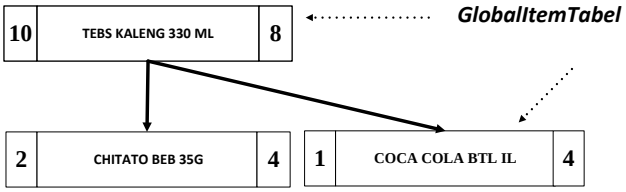
No.	*Path*
5	1
6	1

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 10 dapat dilihat pada gambar 3.20 di bawah:



Gambar 3. 20 *Local CFP-Tree* untuk *item* 10

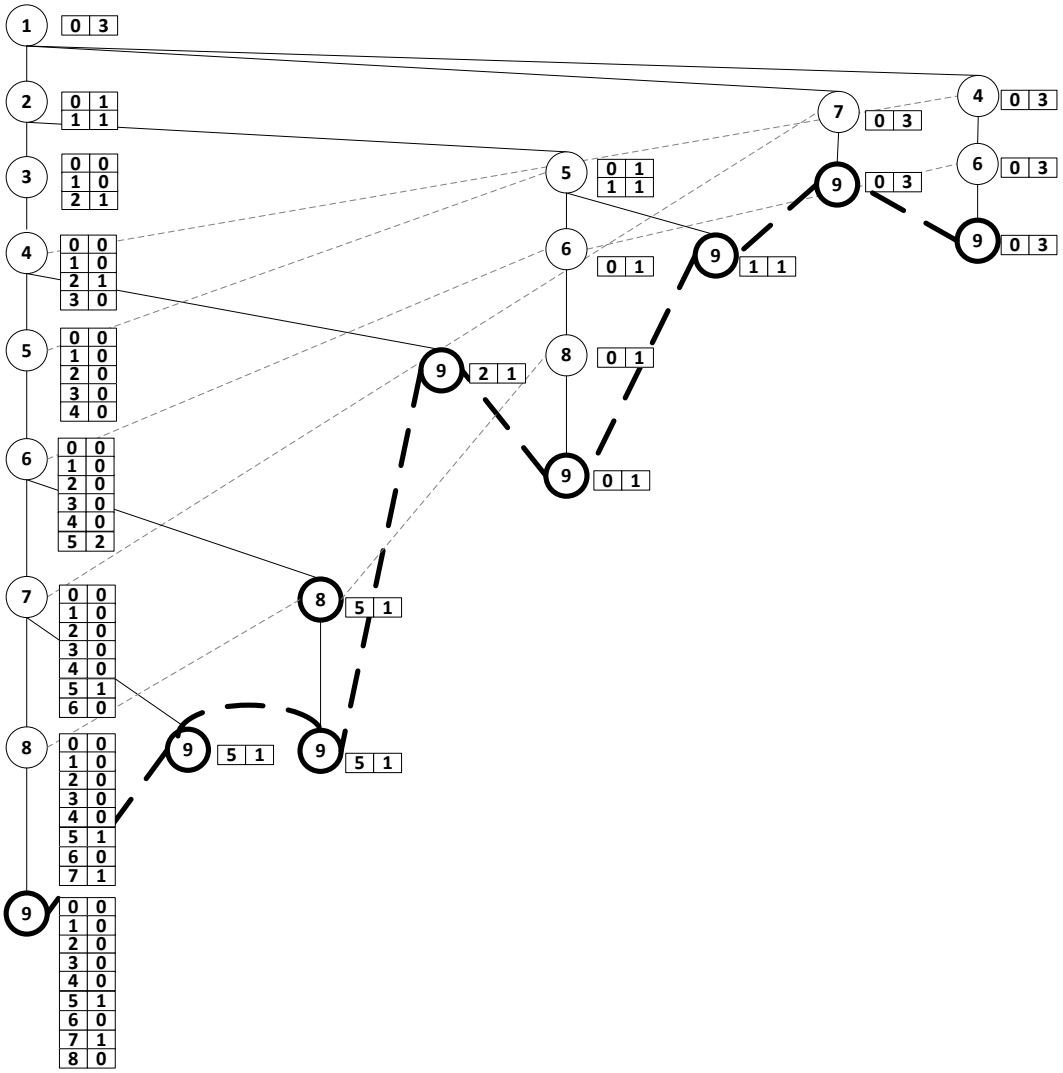
Setelah *Local CFP-Tree index* 10 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 10 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 10. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 10 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.21 di bawah:



Gambar 3. 21 *Local CFP-Tree projection* 10

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 10 dengan nama TEBS KALENG 350 ML adalah sebagai berikut: (10,2:4), (10,1:4).

- Untuk pencarian *local Frequent Pattern Tree index* 9. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 9 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.22 di bawah:



Gambar 3. 22 *CFP-Tree* berakhiran *index* 9

Berikut hasil penelusuran untuk *index* 14 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.25 di bawah:

Tabel 3. 25 Hasil penelusuran *index* 9

No.	*Path*
1	9,8
2	9,7,6

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(3),2(2),3(2),4(2),5(2),6(4),7(2),8(2), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(3),6(4). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk:

Tabel 3. 26 *Local item tabel*

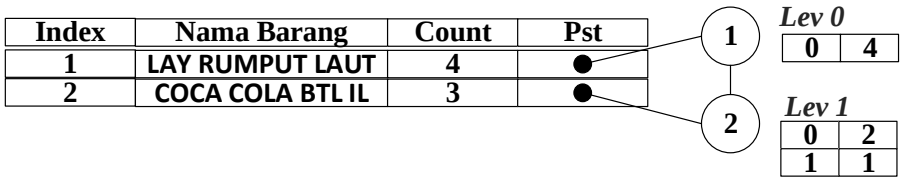
Index local	*Index global*	Nama Barang	*Count*
1	6	LAY RUMPUT LAUT	4
2	2	COCA COLA BTL IL	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat tabel 3.27 di bawah:

Tabel 3. 27 Hasil *Mapping index local*

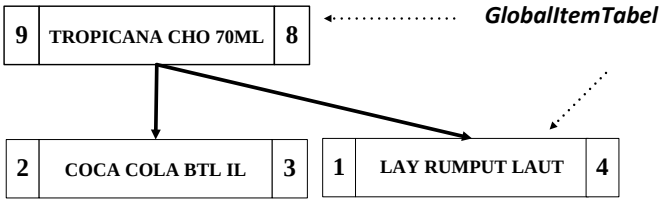
No.	*Path*
1	1
2	1
3	1,2
4	2
5	1,2

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 9 dapat dilihat pada gambar 3.23 di bawah:



Gambar 3. 23 *Local CFP-Tree index* 9

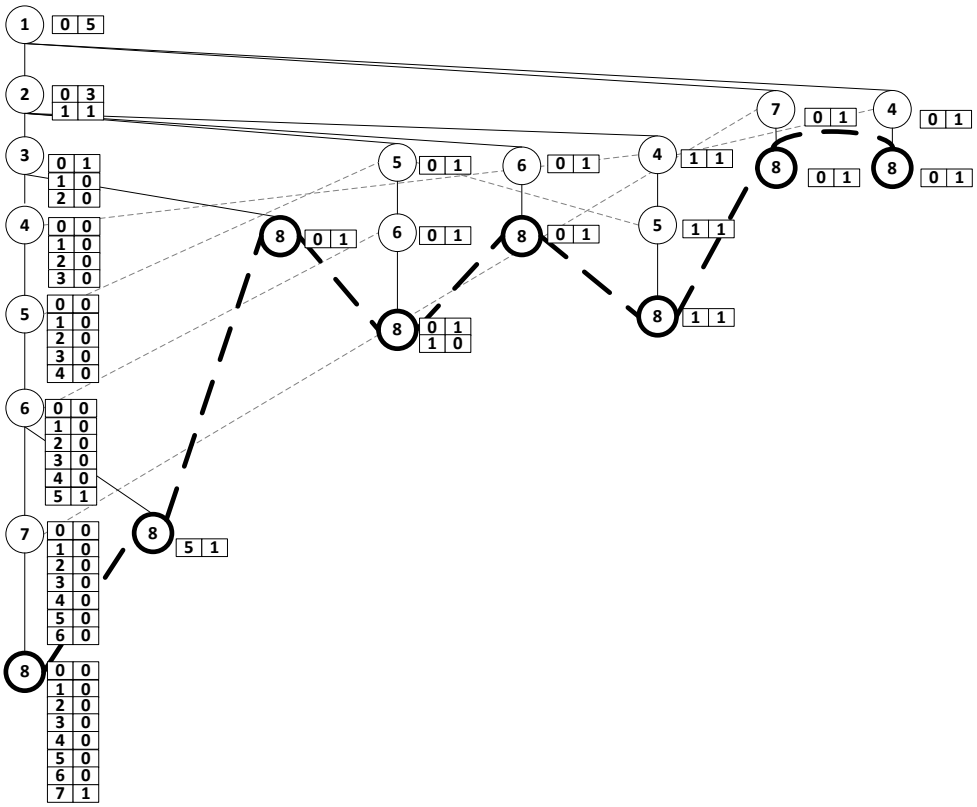
Setelah *Local CFP-Tree index* 9 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 9 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 9. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 9 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.24 di bawah:



Gambar 3. 24 *Local CFP-Tree projection* 9

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 9 dengan nama TROPICANA CHO 70ML adalah sebagai berikut: (9,2:3), (9,1:4).

- Untuk pencarian *local Frequent Pattern Tree index* 8. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 8 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.25 berikut:



Gambar 3. 25 *CFP-Tree* berakhiran *index* 8

Berikut hasil penelusuran untuk *index* 8 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.28 di bawah:

Tabel 3. 28 Hasil penelusuran *index* 8

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

Dari hasil penelusuran atas didapat jumlah *support count* dari setiap *node* yaitu 1(5),2(4),3(1),4(2),5(1),6(3),7(1), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(5),2(4),6(3). Selanjutnya *node* yang

memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.29 di bawah:

Tabel 3. 29 *Local item tabel*

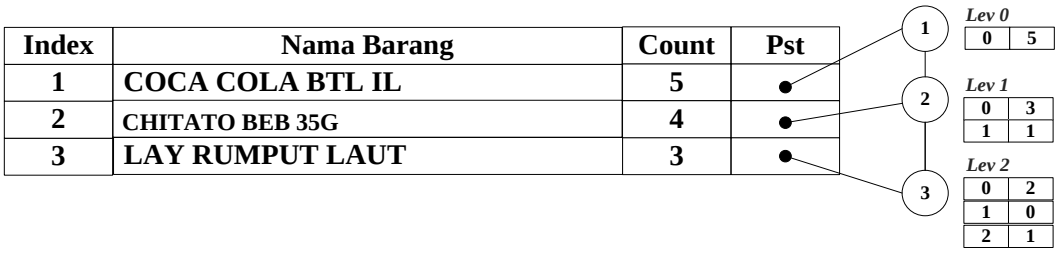
Index local	*Index global*	Nama Barang	*Count*
1	1	COCA COLA BTL IL	5
2	2	CHITATO BEB 35G	4
3	6	LAY RUMPUT LAUT	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.30 di bawah:

Tabel 3. 30 Hasil *Mapping index local*

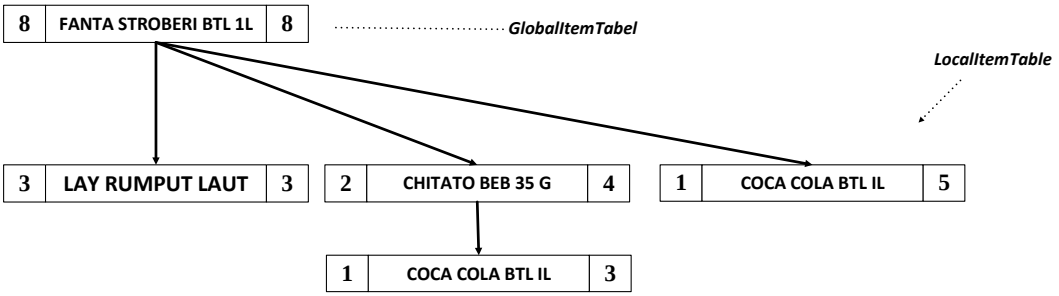
No.	*Path*
1	3
2	1,2
3	1,2,3
4	1,2,3
5	2
6	1
7	1

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 8 dapat dilihat pada gambar 3.26 berikut:



Gambar 3. 26 *Local CFP-Tree index* 8

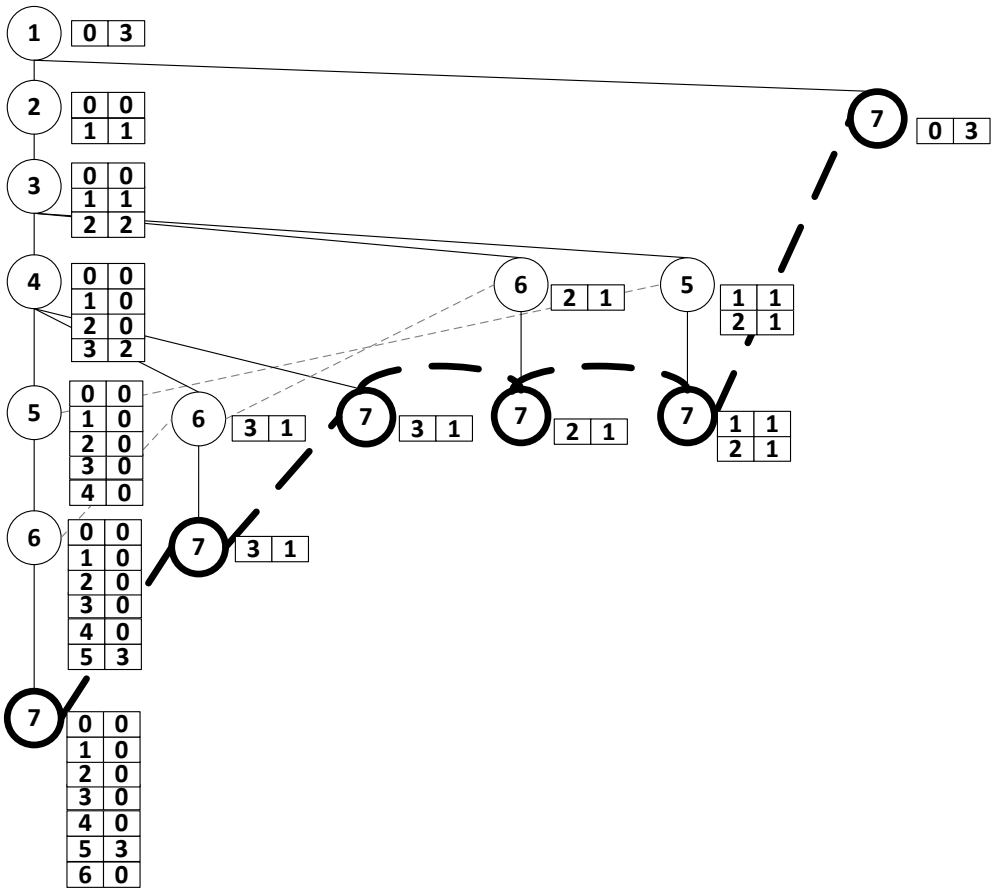
Setelah *Local CFP-Tree index index* 8 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 8 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 8. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 8 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.27 di bawah:



Gambar 3. 27 *Local CFP-Tree projection* 8

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 8 dengan nama Fanta Stroberi BTL 1L adalah sebagai berikut: (8,3:3), (8,2:4),(8,2,1:3),(8,1:5).

- Untuk pencarian *local Frequent Pattern Tree index* 7. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 7 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.28 berikut:



Gambar 3. 28 *CFP-Tree* berakhiran *index* 7

Berikut hasil penelusuran untuk *index* 7 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.31 di bawah:

Tabel 3. 31 Hasil penelusuran *index* 7

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(3),2(1),3(3),4(2),5(2),6(5), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(3), 3(3),6(5). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk pada tabel 3.32 di bawah:

Tabel 3. 32 *Local item tabel*

Index local	*Index global*	Nama Barang	*Count*
1	6	LAY RUMPUT LAUT	5
2	1	COCA COLA BTL IL	3
3	3	SILVER QUEEN MIDI CASH 33G	3

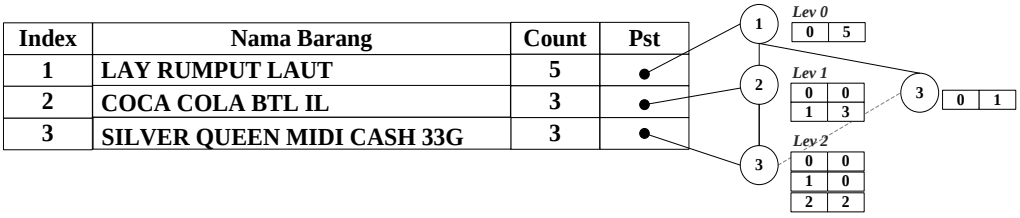
Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.33 di bawah:

Tabel 3. 33 Hasil *Mapping index local*

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

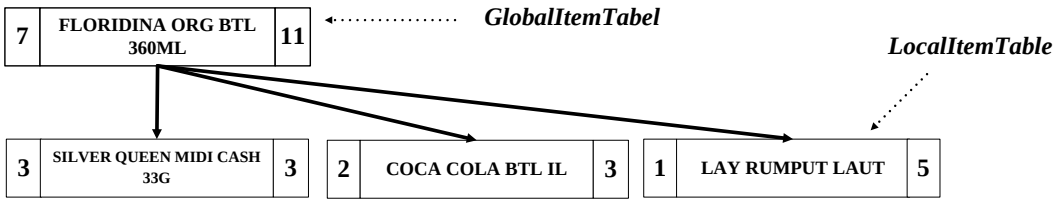
Setelah dilakukan maping data dibuat *Local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang

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 index* 7 dapat dilihat pada gambar 3.29 di bawah:



Gambar 3. 29 *Local CFP-Tree index* 7

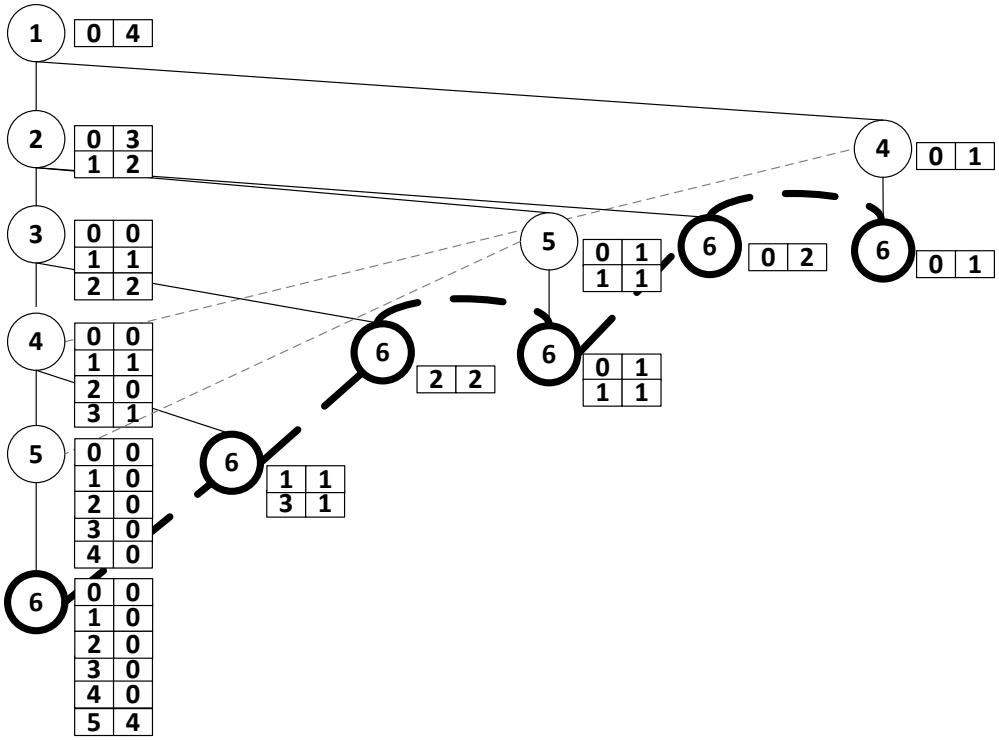
Setelah *Local CFP-Tree index* 7 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 7 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 7. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 7 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.30 di bawah:



Gambar 3. 30 *Local CFP-Tree projection* 7

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 7 dengan nama FLORIDINA ORG BTL 360ML adalah sebagai berikut: (7,3:3), (7,2:3), (7,1:5).

- Untuk pencarian *local Frequent Pattern Tree index* 6. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 6 secara *bottom-up* yaitu penelusuran mulai dari *Path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.31 berikut:



Gambar 3.31 *CFP-Tree* berakhiran *index* 6

Berikut hasil penelusuran untuk *index* 6 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.34 di bawah:

Tabel 3. 34 Hasil penelusuran *index* 6

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(4),2(5),3(3),4(2),5(2), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(4),2(5),3(4). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *local item table* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk pada tabel 3.35 di bawah:

Tabel 3. 35 *Local item tabel*

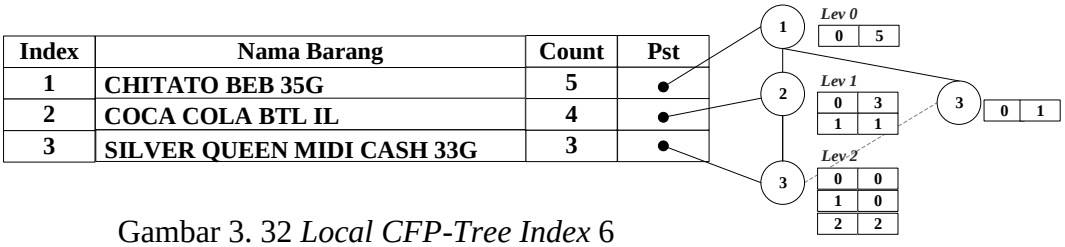
Index local	*Index global*	Nama Barang	*Count*
1	2	CHITATO BEB 35G	5
2	1	COCA COLA BTL IL	4
3	3	SILVER QUEEN MIDI CASH 33G	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.36 di bawah:

Tabel 3. 36 Hasil *Mapping index local*

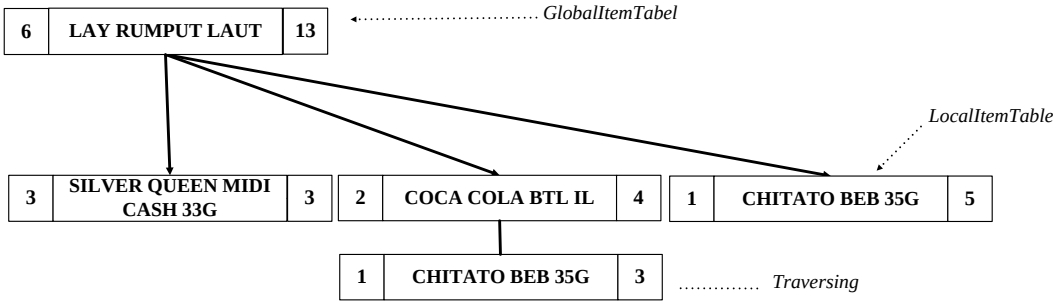
No.	*Path*
1	1,3
2	3
3	3
4	1
5	1,2
6	1.2
7	1.2
8	2

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 6 dapat dilihat pada gambar 3.32 di bawah:



Gambar 3. 32 *Local CFP-Tree Index* 6

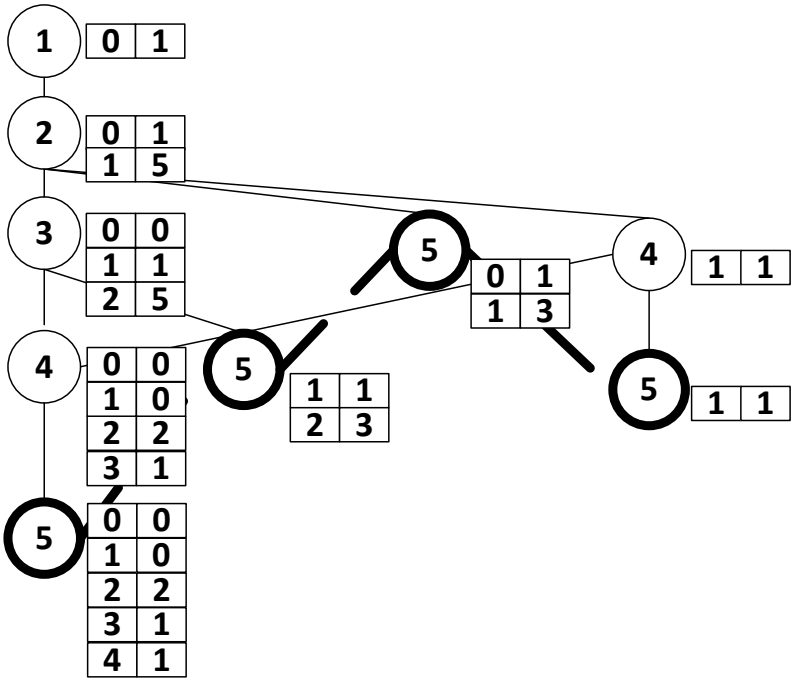
Setelah *Local CFP-Tree index* 6 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 6 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 6. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 6 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.33 di bawah:



Gambar 3. 33 *Local CFP-Tree projection* 6

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 6 dengan nama LAY RUMPUT LAUT adalah sebagai berikut: (6,3:3), (6,2:4), (6,2,1:3) (6,1:5).

- Untuk pencarian *local Frequent Pattern Tree index* 5. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 5 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.34 di bawah:



Gambar 3. 34 *CFP-Tree* berakhiran *index* 5

Berikut hasil penelusuran untuk *index* 5 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.37 di bawah:

Tabel 3. 37 Hasil penelusuran *index* 5

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

No.	*Path*
10	5,2
11	5,2
12	5,2,1
13	5,4,2,1

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(1),2(6),3(6),4(3), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 2(6),3(5),4(4). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.38 di bawah:

Tabel 3. 38 *Local item tabel*

Index local	*Index global*	Nama Barang	*Count*
1	2	CHITATO BEB 35G	6
2	3	SILVER QUEEN MIDI CASH 33G	6
3	4	QTELA KEJU PANGGANG 60G	4

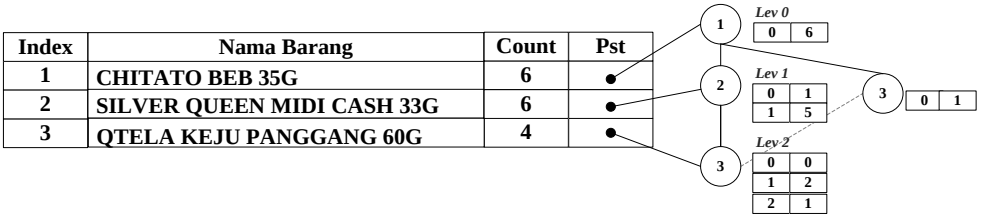
Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.39 di bawah:

Tabel 3. 39 Hasil *Mapping index local*

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

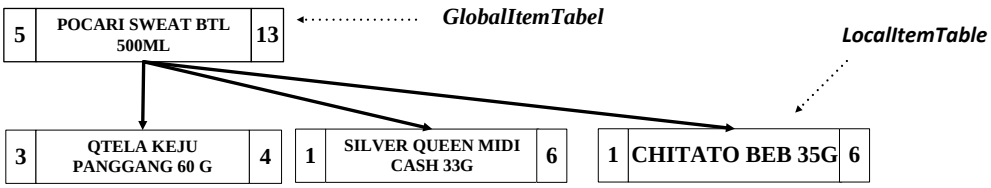
No.	*Path*
10	1
11	1
12	1,3

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 5 dapat dilihat pada gambar 3.35 di bawah:



Gambar 3. 35 *Local CFP-Tree index* 5

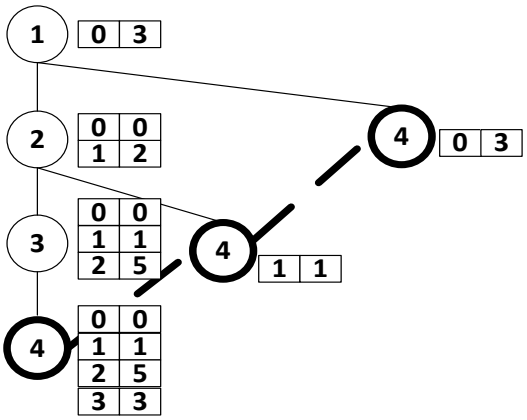
Setelah *Local CFP-Tree index* 5 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 5 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 5. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 5 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.36 di bawah:



Gambar 3. 36 *Local CFP-Tree projection* 5

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 5 dengan nama POCARI SWEAT BTL 500ML adalah sebagai berikut: (5,3:3), (5,2:6), (5,1:6).

- Untuk pencarian *local Frequent Pattern Tree index* 4. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 4 secara *bottom-up* yaitu penelusuran mulai dari *path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.37 di bawah:



Gambar 3. 37 *CFP-Tree* berakhiran *index* 4

Berikut hasil penelusuran untuk *index* 4 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.40 di bawah:

Tabel 3. 40 Hasil penelusuran *index* 4

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

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(3),2(2),3(5), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(3),3(6). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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:

Tabel 3. 41 *Local item tabel*

Index local	*Index global*	Nama Barang	*Count*
1	1	SILVER QUEEN MIDI CASH 33G	6
2	3	COCA COLA BTL IL	3

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.42 di bawah:

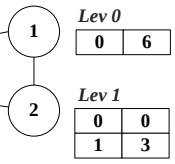
Tabel 3. 42 Hasil *Mapping index local*

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

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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

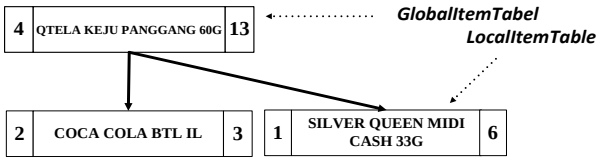
Global Item tabel yang terbentuk dari *Global Item tabel* data sedangkan pada *Local CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Local item tabel* yang terbentuk dari *Local Frequent item*. Berikut adalah *Local CFP-Tree index* 4 dapat dilihat pada gambar 3.38 di bawah:

Index	Nama Barang	Count	Pst
1	SILVER QUEEN MIDI CASH 33G	6	
2	COCA COLA BTL 1L	3	



Gambar 3. 38 *Local CFP-Tree index* 4

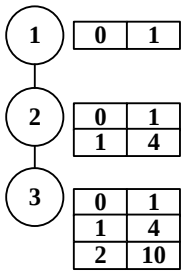
Setelah *Local CFP-Tree index* 4 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 4 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 4. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 4 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.39 dii bawah:



Gambar 3. 39 *Local CFP-Tree projection* 4

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 4 dengan nama QTELA KEJU PANGGANG 60G adalah sebagai berikut: (4,2:3), (4,1:6).

- Untuk pencarian *local Frequent Pattern Tree index* 3. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 3 secara *bottom-up* yaitu penelusuran mulai dari *Path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.40 berikut:



Gambar 3. 40 *CFP-Tree* berakhiran *index* 3

Berikut hasil penelusuran untuk *index* 4 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.43 di bawah:

Tabel 3. 43 Hasil penelusuran *index* 3

No.	*Path*
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3,2
12	3,2
13	3,2
14	3,2
15	3,2,1

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(1),2(5), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 2(5). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.44:

Tabel 3. 44 *Local item tabel*

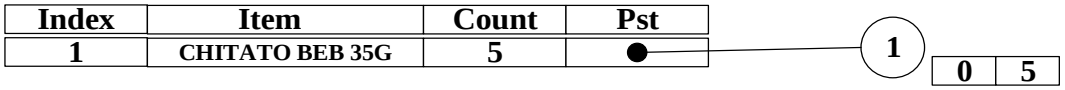
Index local	*Index global*	Nama Barang	*Count*
1	2	CHITATO BEB 35G	5

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.45 di bawah:

Tabel 3. 45 Hasil *Mapping index local*

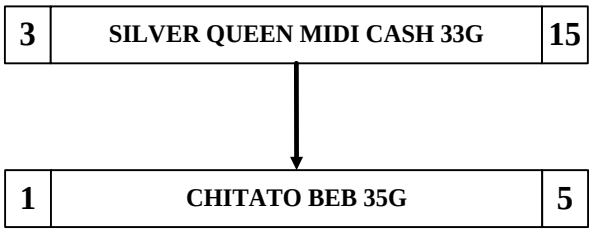
No.	*Path*
1	1
2	1
3	1
4	1
5	1

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 3 dapat dilihat pada gambar 3.41 di bawah:



Gambar 3. 41 *Local CFP-Tree index* 3

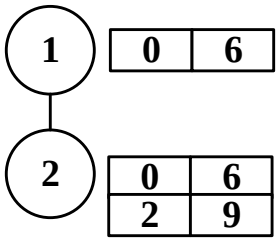
Setelah *Local CFP-Tree index* 3 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 3 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 3. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 3 mulai dari *path sampai root* secara *bottom-up*, untuk lebih jelasnya dapat dilihat pada gambar 3.42 berikut:



Gambar 3. 42 *Local CFP-Tree projection* 3

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 3 dengan nama SILVER QUEEN MIDI CASH 33G adalah sebagai berikut:(2,1:5).

- Untuk pencarian *local Frequent Pattern Tree index* 2. Lakukan penelusuran pada *Global CFP-Tree* yang hanya berakhiran di *index* 2 secara *bottom-up* yaitu penelusuran mulai dari *Path* sampai ke *root*, untuk jelasnya dapat dilihat pada gambar 3.43 di bawah:



Gambar 3. 43 *CFP-Tree* berakhiran *index* 2

Berikut hasil penelusuran untuk *index* 4 pada *Global CFP-Tree* yang dapat dilihat di bawah:

Tabel 3. 46 Hasil penelusuran *index* 2

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

No.	*Path*
11	2,1
12	2,1
13	2,1
14	2,1
14	2,1

Dari hasil penelusuran didapat jumlah *support count* dari setiap *node* yaitu 1(6), karena *minimum support* nya 3, maka *node* yang memenuhi *minimum support* adalah 1(6). Selanjutnya *node* yang memenuhi *minimum support* dimasukkan ke *local item table dan* diberikan *index* baru di *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 3.47:

Tabel 3. 47 *Local item tabel*

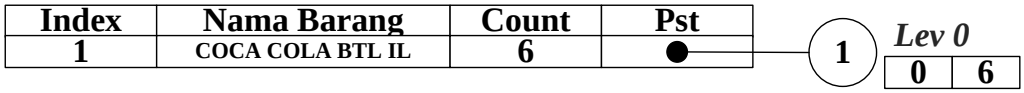
Index local	*Index global*	*Nama barang*	*Count*
1	1	COCA COLA BTL IL	6

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dengan nomor *index* yang sudah diberikan. Berikut hasil *Mapping* yang dapat dilihat pada tabel 3.48 di bawah:

Tabel 3. 48 Hasil *Mapping index local*

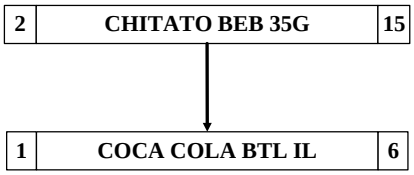
No.	*Path*
1	1
2	1
3	1
4	1
5	1
6	1

Setelah dilakukan maping data dibuat *Local CFP-Tree*. 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 index* 2 dapat dilihat pada gambar 3.44 di bawah:



Gambar 3. 44 *Local CFP-Tree index* 2

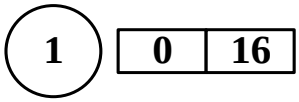
Setelah *Local CFP-Tree index* 2 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* di mana *index* 2 akan menjadi *root* dan setiap *index* di *Local CFP-Tree* akan menjadi anak dari *index* 2. Untuk menemukan *local frequent pattern tree*, lakukan penelusuran di *Local CFP-Tree index* 2 mulai dari *path sampai root* secara *bottom-up,* untuk lebih jelasnya dapat dilihat pada gambar 3.45 di bawah:



Gambar 3. 45 *Local CFP-Tree projection* 2

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 2 dengan nama CHITATO BEB 35G adalah sebagai berikut: (2,1:6).

- Untuk pencarian *local Frequent Pattern Tree* untuk *item* 1. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran di *item* 1, untuk jelasnya dapat dilihat pada gambar 3.46 di bawah:



Gambar 3. 46 *CFP-Tree* berakhiran *index* 1

Dari gambar 3.46 di atas tidak ada *node* yang ditemukan, maka tidak terbentuk *Local item tabel* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

Dari *frequent Pattern* yang didapat, akan di-*generate* untuk mendapatkan *rule* dengan *minimum confidence* = 60%. Dihitung dengan persamaan 2-4 sebagai berikut

$$Confidence = P(B \mid A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ transaksi\ yang\ mengandung\ A}\ X\ 100\%$$

Persamaan (2-1)

Berikut adalah hasil *generate rule* dapat dilihat pada tabel 3.49 di bawah:

Tabel 3. 49 Hasil *Generate Rule*

Frequent itemset	*Confidence*
14 ➔ 5	3/7 * 100% = 42%
14 ➔ 4	3/7 * 100% = 42%
14 ➔ 3	3/7 * 100% = 42%
14 ➔ 1	5/7 * 100% = 71%
13 ➔ 4	3/8 * 100% = 37%
13 ➔ 3	3/8* 100% = 37%
13 ➔ 2	3/8 * 100% = 37%
13 ➔ 2,1	3/8 * 100% = 37%
13 ➔ 1	4/8 * 100% = 50%
12➔4	3/8 * 100% = 37%
12➔3	4/8 * 100% = 50%
12 ➔ 2	5/8 * 100% = 62%
12 ➔2,1	3/8 * 100% = 37%
12 ➔ 1	5/8 * 100% = 62%
11 ➔ 3	4/8 * 100% = 50%
11 ➔ 3,1	3/8 * 100% = 37%
11 ➔ 2	4/8 * 100% = 50%

Frequent itemset	*Confidence*
11 ➔ 1	4/8 * 100% = 50%
10➔2	4/8 * 100% = 50%
10➔1	4/8 * 100% = 50%
9➔2	3/8 * 100% = 37%
9➔1	4/8 * 100% = 50%
8➔3	3/8 * 100% = 37%
8➔2	4/8 * 100% = 50%
8➔2,1	3/8 * 100% = 37%
8➔1	5/8 * 100% = 62%
7➔3	3/11 * 100% = 27%
7➔2	3/11 * 100% = 27%
7➔1	5/11 * 100% = 45%
6➔3	3/13 * 100% = 23%
6➔2	4/13 * 100% = 30%
6➔2,1	3/13 * 100% = 23%
6➔1	5/13 * 100% = 38%
5➔3	4/13 * 100% = 30%
5➔2	6/13 * 100% = 43%
5➔1	6/13 * 100% = 43%
4➔2	3/13 * 100% = 23%
4➔1	6/13 * 100% = 43%
3➔1	5/15 * 100% = 33%
2➔1	6/15 * 100% = 40%

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 dari masing-masing *id* agar pembacaan *rule* lebih jelas. Berikut hasil penentuan paket barang dapat dilihat pada tabel 3.50:

Tabel 3. 50 Hasil *Rule* yang terbentuk

Rules	Paket Barang	*Confidence*
14 → 1	LUWAK WHITE KOFFIE 20G→COCA COLA BTL IL	71%
12 → 2	CHITATO RASA AYAM BEB 35G →CHITATO ORI FLAVOR 35 G	62%
12 → 1	CHITATO RASA AYAM BEB 35G → COCA COLA BTL IL	62%
8→1	FANTA STROBERI BTL 1L → COCA COLA BTL IL	62%

Setalah mengeliminasi *rule* yang tidak memenuhi *minimum Confidence* tahap selanjutnya adalah membentuk paket barang dalam bentuk awal dengan menjumlahkan setiap harga barang dalam satu *rule*. Berikut merupakan hasil pembentukan paket barang dalam bentuk awal dapat dilihat pada tabel 3.51:

Tabel 3. 51 Paket barang dalam bentuk awal

*Rule*s	Paket Barang	Harga
14 →1	LUWAK WHITE KOFFIE 20G→COCA COLA BTL IL	18,500
12 →2	CHITATO RASA AYAM BEB 35G →CHITATO ORI FLAVOR 35 G	10,000
12 →1	CHITATO RASA AYAM BEB 35G → COCA COLA BTL IL	13,500
8→1	FANTA STROBERI BTL 1L → COCA COLA BTL IL	17,500

Dalam penentuan kombinasi yang akan ditawarkan dalam bentuk paket, Mini Market HD memiliki batasan khusus, yaitu:

1. Jumlah barang yang dipaketkan yaitu sebanyak 2 sampai 7 barang.
2. Harga paket minimal sebesar Rp.15.000 dan maksimal Rp.70.000.

Dari batasan paket yang ditentukan oleh Mini Market Hd Mart, maka didapat rekomendasi paket dapat dilihat pada tabel 3.52 di bawah:

Tabel 3. 52 Paket yang terbentuk

No.	Paket Barang	Harga
Paket 1	1. LUWAK WHITE KOFFIE 20G 2. COCA COLA BTL IL	18,500
Paket 2	1. FANTA STROBERI BTL 1L 2. COCA COLA BTL IL	17,500

3.1.3 Analisis Kebutuhan Non Fungsional

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

6.1.3.1 Analisis Pengguna

User atau pengguna yang akan mengoperasikan aplikasi ini merupakan pemilik yang bertindak sebagai pengolah bisnis. Analisis Pengguna dapat dilihat pada tabel 3.53.

Tabel 3. 53 Analisis Pengguna

User	Akses	Keahlian
Pemilik	Pemilik dapat mengolah data, dapat mengoperasikan aplikasi dengan baik seperti *import* data transaksi dan mengolah informasi yang didapat memperoleh informasi dari aplikasi *mining*.	Dapat mengoperasikan aplikasi dengan baik seperti mengimput 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 di bagian pemilik Mini Market HD Mart.

1. Analisis spesifikasi perangkat lunak pada sistem yang sedang berjalan di bagian pemilik Mini Market HD Mart 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 pemilik Mini Market HD Mart, spesifikasi perangkat lunak yang digunakan masih kurang. Untuk mengatasi hal tersebut akan di *instal 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 pemilik Mini Market HD Mart.

1. Analisis spesifikasi perangkat keras pada *system* yang sedang berjalan di bagian pemilik Mini Market HD Mart adalah:
   a. *Processor* : *Intel Core* i3-550 3.2 *Ghz*
   b. *Hardisk* : 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. *Hardisk* : 500 MB untuk menyimpan *File*
   c. *Memory* : 1 GB
3. Evaluasi kebutuhan perangkat lunak

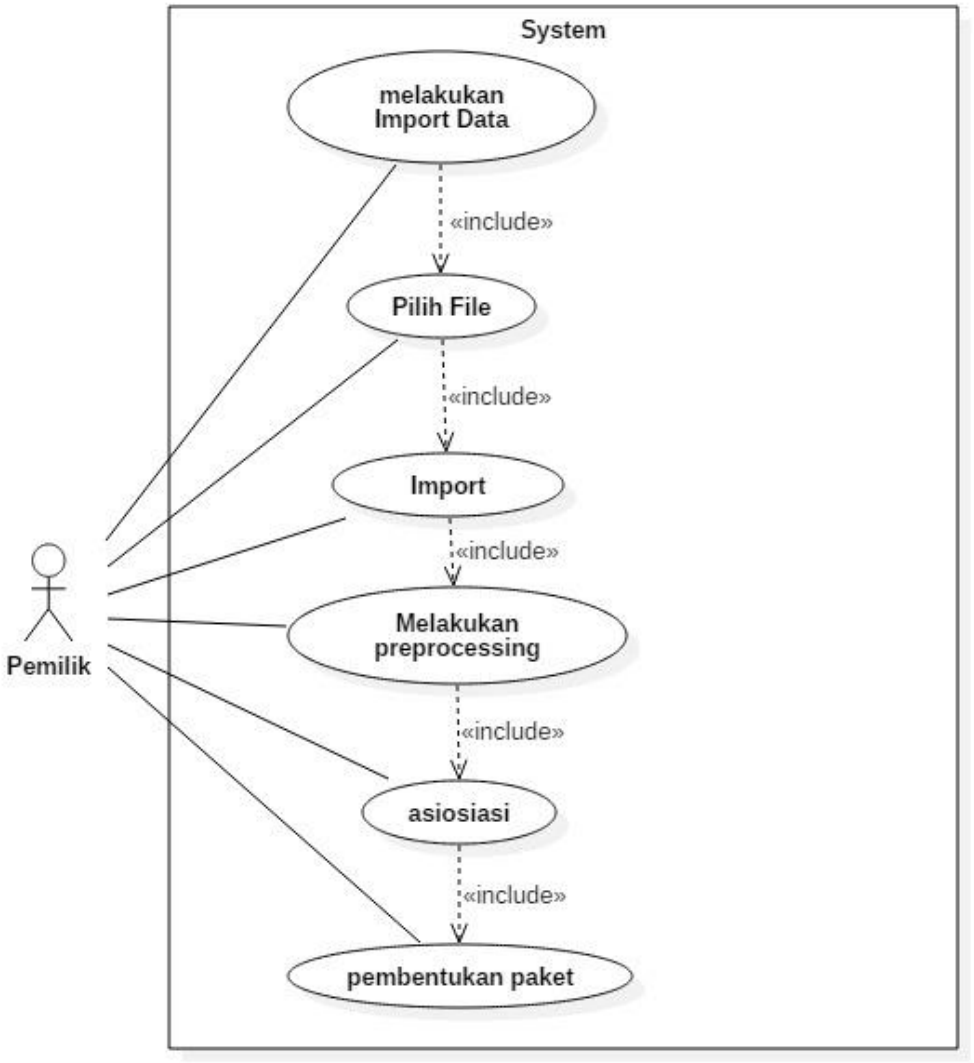
Berdasarkan analisis spesifikasi perangkat keras yang ada di bagian pemilik Mini Market HD Mart, perangkat keras yang digunakan sudah cukup untuk dapat menjalankan aplikasi *Data Mining*.

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 digambarkan dengan analisis Pemrograman Berorientasi Objek dengan menggunakan konsep *UML* (*Unified Modeling Languages*).

3.1.4.1. Diagram *Use Case*

Use case atau Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* dapat dilihat pada gambar 3.47 dan Penjelasan *user* dan *use case* dapat diacu pada tabel 3.54 dan tabel deskripsi *use case* yang diacu pada tabel 3.55.



Gambar 3. 47 Diagram *Use Case*

Tabel 3. 54 Tabel Definisi *User*

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

Tabel 3. 55 Tabel Deskripsi Use Case

No	*Use Case*	Deskripsi
1	*Melakukan Import* Data	*Import* data harus dilakukan user terlebih dahulu, data yang di *import* merupakan data transaksi yang selanjutnya data tersebut akan disimpan ke dalam *database*
2	Pilih *File*	Pemilik memilih data transaksi
3	*Import*	Pemilik meng*import* data transaksi ke dalam *database*
4	*Melakukan Preprocessing*	Data transaksi yang sudah tersimpan dalam *database* kemudian akan dilakukan proses pembersihan data, pemilihan atribut, dan hitung frekuensi
5	Asosiasi	Pemilik menginputkan nilai parameter yaitu nilai *Minimum support* dan *Minimum Confidence*, kemudian sistem melakukan pencarian *frequent itemset,* melakukan proses eliminasi, menghasilkan *rule* dan informasi rekomendasi paket.
6	Pembentukan paket	Sistem menampilkan informasi berupa barang apa saja yang bisa dipasangkan dalam bentuk paket.

3.1.4.1.1 Skenario *Use Case*

1. Skenario *Use Case* Melakukan *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*.

Requirement A.1

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

Tabel 3. 56 Skenario Use Case Melakukan *Import* Data

Use case Name	*Import Data*	
Related Requirements	*Requirement A.1*	
Goal in Context	*Import* data transaksi HD Mart dan menyimpannya ke dalam *database*	
Precondition	Pemilik 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	Pemilik	
Triger	Pemilik memasukkan data transaksi	
Included Cases	-	
Main Flow	Step	*Action*
	1	Pemilik menekan tombol *import* data
	2	Sistem menampilkan menu *import* data

	3	Pemilik menekan tombol pilih *file*
	4	Pemilik memilih data transaksi
	5	sistem menampilkan hasil ekstraksi data transaksi
	6	Pemilik menekan tombol *import*
	7	Sistem menyimpan data transaksi ke dalam *database*
	8	menampilkan pesan data berhasil disimpan
Extension	Step	Branching *Action*
	1.1	Pemilik memilih membatalkan proses *import*.
	2.1	Menampilkan pesan kesalahan dalam proses *import* data

2. Skenario *Use Case* Pilih *File*

Skenario *use case* pilih *File* menggambarkan langkah-langkah aksi *user* untuk melakukan proses pilih *File*.

Tabel 3. 57 Skenario Use Case Pilih *File*

Use case Name	Pilih *File*	
Related Requirements	*Requirement A.1*	
Goal in Context	Sistem berhasil mengekstrak data transaksi yang digunakan	
Precondition	*System* menampilkan menu *import* data	
Successful End Condition	Sistem menampilkan data transaksi berhasil menampilkan data transaksi	
Failed End Condition	Sistem gagal menampilkan data transaksi	
Actors	Pemilik	
Triger	Pemilik memilih data transaksi yang akan digunakan	
Include case	*Import* data	
Main Flow	Step	*Action*
	1	Pemilik menekan tombol pilih *file*
	2	Pemilik memilih *file* transaksi
	3	Sistem mengekstrak data transaksi
	4	Sistem menampilkan data transaksi
Extension	Step	Branching *Action*
	1	Pemilik tidak memilih data transaksi

3. Skenario Use Case *Import*

Skenario *use case* pilih *File* menggambarkan langkah-langkah aksi *user* untuk melakukan proses *import*

Tabel 3. 58 Skenario Use Case Pilih *File*

Use case Name	Pilih *File*
Related Requirements	*Requirement A.1*

Goal in Context	Sistem berhasil *import* data transaksi kedalam data base	
Precondition	Pemilik telah memilih data transaksi	
Successful End Condition	Data transaksi berhasil di impor kedalam *database*	
Failed End Condition	Sistem gagal menyimpan data transaksi	
Actors	Pemilik	
Triger	Pemilik memilih data transaksi yang akan digunakan	
Include case	*Import* data, pilih *File*	
Main Flow	Step	*Action*
	1	Pemilik menekan tombol *import*
	2	Sistem menyimpan data transaksi ke *database*
	3	Sistem menampilkan pesan berhasil di*import*
	4	Sistem menampilkan data transaksi
Extension	Step	Branching *Action*
	1	Pemilik tidak memilih data transaksi

4. Skenario *Use Case* Melakukan *Preprocesing*

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

Requirement A.2

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

Tabel 3. 59 Skenario Use Case Melakukan Preprocesing

Use case Name	*Preprocesing*
Related Requirements	*Requirement A.1, Requirement A.2*
Goal in Context	Sistem melakukan proses pemilihan atribut, pembersihan data, dan *support count*

Precondition	Data transaksi sudah tersimpan di dalam *database*	
Successful End Condition	Sistem berhasil melakukan proses pemilihan atribut, pembersihan data, dan *support count*	
Failed End Condition	Sistem gagal untuk melakukan proses pemilihan atribut, pembersihan data dan *support count*	
Actors	Pemilik	
Triger	Pemilik melakukan *request* proses pemilihan atribut, pembersihan data, dan *support count*	
Included Cases	*Import* data	
Main Flow	Step	*Action*
	1	Pemilik memilih tombol *Preprocessing*
	2	Sistem menampilkan menu *Preprocessing*
	3	Pemilik menekan tombol proses
	4	Sistem melakukan pemilihan atribut
	5	Sistem menyimpan hasil pemilihan atribut
	6	Sistem menampilkan pemilihan atribut
	7	Sistem melakukan pembersihan data transaksi tunggal
	8	Sistem menampilkan hasil proses Pembersihan Data
	9	Sistem melakukan perhitungan *support count* pada setiap barang
	10	Sistem menampilkan hasil proses Hitung *support count*
	11	Sistem menyimpan hasil perhitungan *support count*
	12	Sistem menyimpan hasil proses *support count* ke dalam *database*
	13	Sistem menampilkan pesan *Preprocessing* berhasil

Extension	Step	Branching *Action*
	4.1	Sistem menampilkan pesan kesalahan gagal dalam proses pemilihan atribut
	8.1	Sistem menampilkan pesan kesalahan gagal dalam proses pembersihan data
	10.1	Sistem menampilkan pesan kesalahan gagal dalam proses hitung *support count*

5. Skenario Use Case Proses Asosiasi

Skenario *use case* proses *asosiasi* menggambarkan langkah-langkah aksi *user* untuk melakukan proses *Input* Parameter, Lihat *Rules*, dan Lihat Informasi Paket.

Requirement A.3

Requirement A.3
Sistem menyediakan menu asosiasi untuk melakukan input parameter berupa *Minimum support* dan *Minimum confidence* untuk mengeliminasi *item* apa saja yang tidak memenuhi nilai parameter yang dimasukan, Lihat *Rule*s untuk melihat *rule* apa saja yang dihasilkan, dan Lihat Informasi paket untuk melihat informasi pembentukan paket yang dihasilkan

Tabel 3. 60 Skenario Use Case Asosiasi

Use case Name	Asosiasi
Related Requirements	*Requirement A.1, Requirement A.2*
Goal in Context	Sistem mendapatkan *rule*
Precondition	Data sudah melalui proses *preprocessing*
Successful End Condition	Sistem berhasil menampilkan *rule*
Failed End Condition	Sistem gagal mendapatkan *rule*
Actors	Pemilik

Triger	Pemilik memasukkan nilai parameter berupa nilai *Minimum support* dan *Minimum confidence*	
Include case	*Preprocessing*	
Main Flow	Step	*Action*
	1	Pemilik memilih menu asosiasi
	2	Sistem menampilkan menu asosiasi
	2	Pemilik memasukkan nilai *Minimum* dan *Minimum Confidence*
	3	Pemilik *menekan tombol Proses*
	3	Sistem melakukan proses pencarian *frequent itemset* sesuai nilai *Minimum support*
	4	Sistem mendapatkan *frequent itemset*
	5	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*
Extension	Step	Branching *Action*
	2.1	*Minimum* dan *Minimum Confidence* tidak boleh kosong
	4.1	Sistem tidak mendapatkan *frequent itemset* karena nilai *Minimum support* yang dimasukkan oleh *user* terlalu tinggi
	5.1	Sistem gagal melakukan *generate rule*, karna tidak ada *frequent itemset* yang ditemukan
	7.1	Sistem tidak mendapatkan *rule* karena nilai *minimum confidence* yang dimasukkan terlalu tinggi

	8.1	Sistem gagal menampilkan hasil dari *generate rule* karena nilai *support* / *confidence*nya terlalu tinggi

6. Skenario *Use Case* Pembentukan Paket

Skenario *use case* pembentukan paket menggambarkan langkah-langkah aksi *user* untuk melakukan proses Input Parameter Harga, Lihat Pembentukan Harga. Skenario *use case* pembentukan harga dapat dilihat pada table 3.68

Tabel 3. 64 *Requirement* A.3

Requirement A.3
Sistem menyediakan pembentukan paket untuk melakukan *input* parameter berupa minimal Harga dan maksimal untuk memfilter *rule* dan membentuk barang apa saja yang memenuhi nilai parameter yang dimasukkan

Tabel 3. 65 Skenario *Use Case* Pembentukan Paket

Use case Name	Asosiasi	
Related Requirements	*Requirement A.1, Requirement A.2, Requirement A.3,*	
Goal in Context	Sistem mendapatkan hasil filter *rule* yang memenuhi nilai parameter yang di masukan	
Precondition	Data sudah melalui proses *Asosiasi*	
Successful End Condition	Sistem berhasil filter *rule* yang memenuhi nilai parameter yang di masukan	
Failed End Condition	Sistem gagal memfilter *rule* yang memenuhi nilai parameter yang di masukan	
Actors	Pemilik	
Triger	Pemilik memasukan nilai parameter berupa nilai minimal harga dan maksimum harga	
Included Cases	*Input* Parameter harga, Lihat Pembentukan Paket	
Main Flow	Step	*Action*

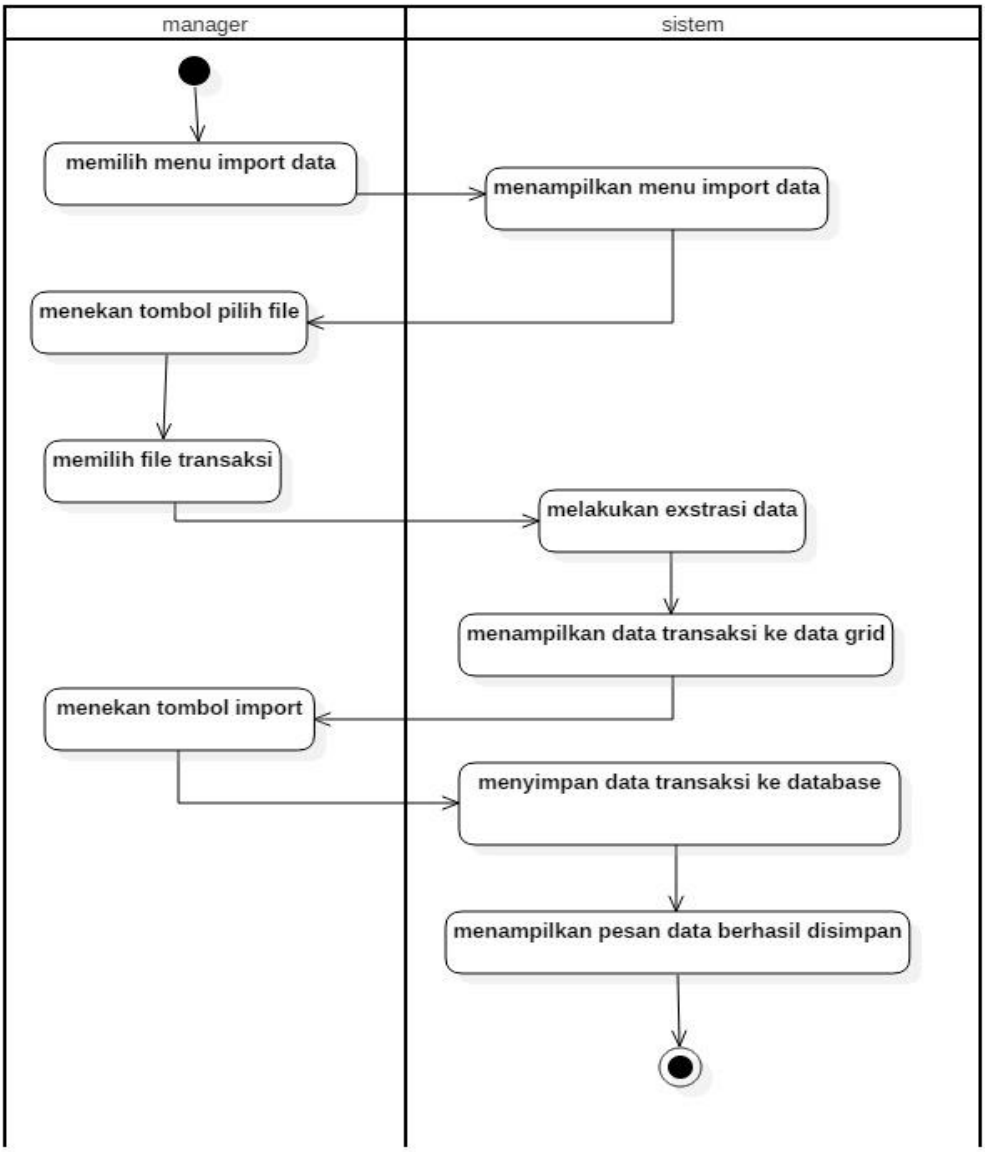
	1	Pemilik memilih menu pembentukan paket
	2	Pemilik memasukkan nilai *minimum* harga dan nilai maksimum harga
	3	Sistem menghitung jumlah harga dari setiap *rule*
	4	Sistem memfilter *rule* yang memenuhi nilai parameter
	5	Sistem menampilkan data paket yang memenuhi nilai parameter
Extension	Step	*Branching Action*
	1.1	Sistem menampilkan notifikasi nilai *minimum* harga yang dimasukkan tidak sesuai
	3.1	Sistem tidak mendapatkan *rule* karena nilai *Minimum* harga dan maksimum harga terlalu tinggi

3.1.4.2 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*

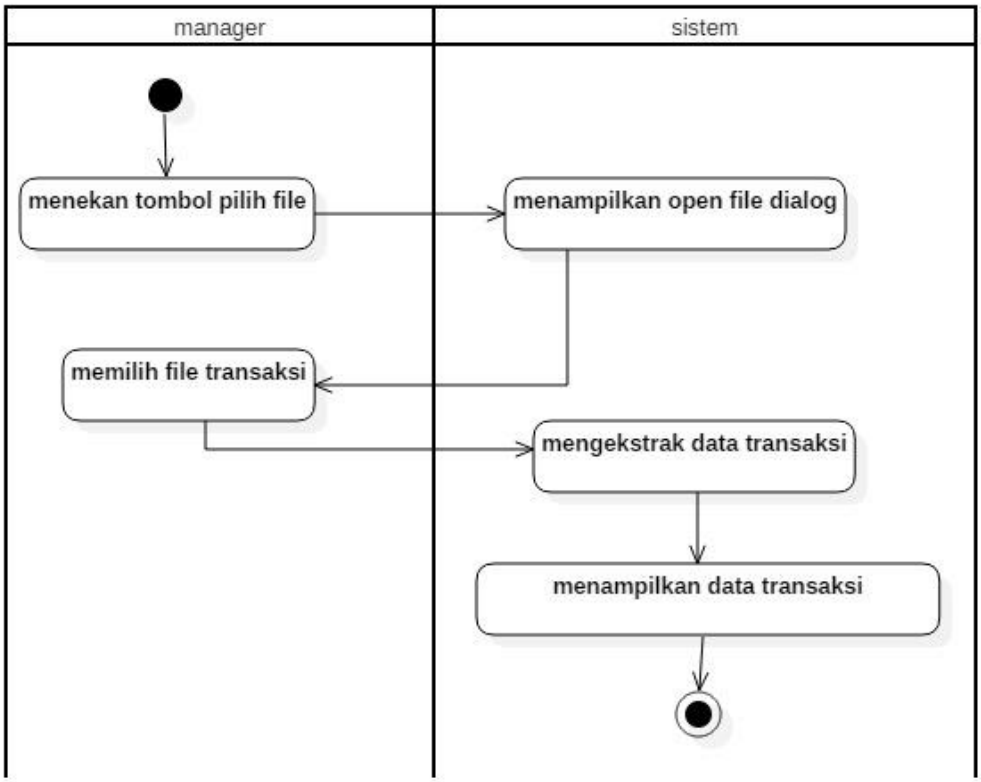
   Activity Diagram *import data* menjelaskan aktivitas aktor dalam melakukan proses *import* data. Penjelasan *activity* Diagram *import* data diacu pada Gambar 3.48:



Gambar 3. 48 *Activity* Diagram *Import* Data

2. *Activity* Diagram Pilih *File*

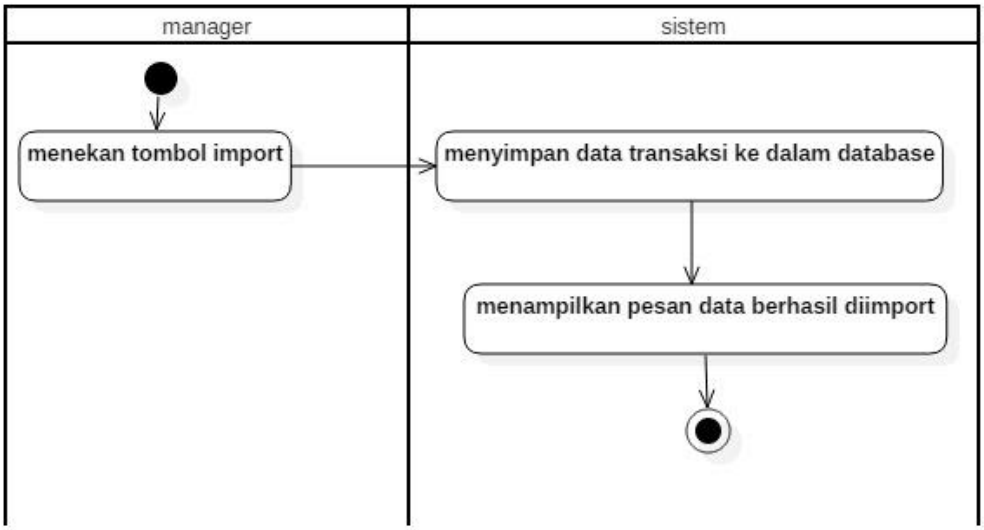
Activity Diagram pilih *File* menjelaskan aktivitas sistem dalam melakukan proses memilih data transaksi. *Activity* Diagram pilih *File* dapat dilihat pada gambar 3.49 di bawah ini:



Gambar 3. 49 *Activity* Diagram Pilih *File*

3. *Activity* Diagram *Import*

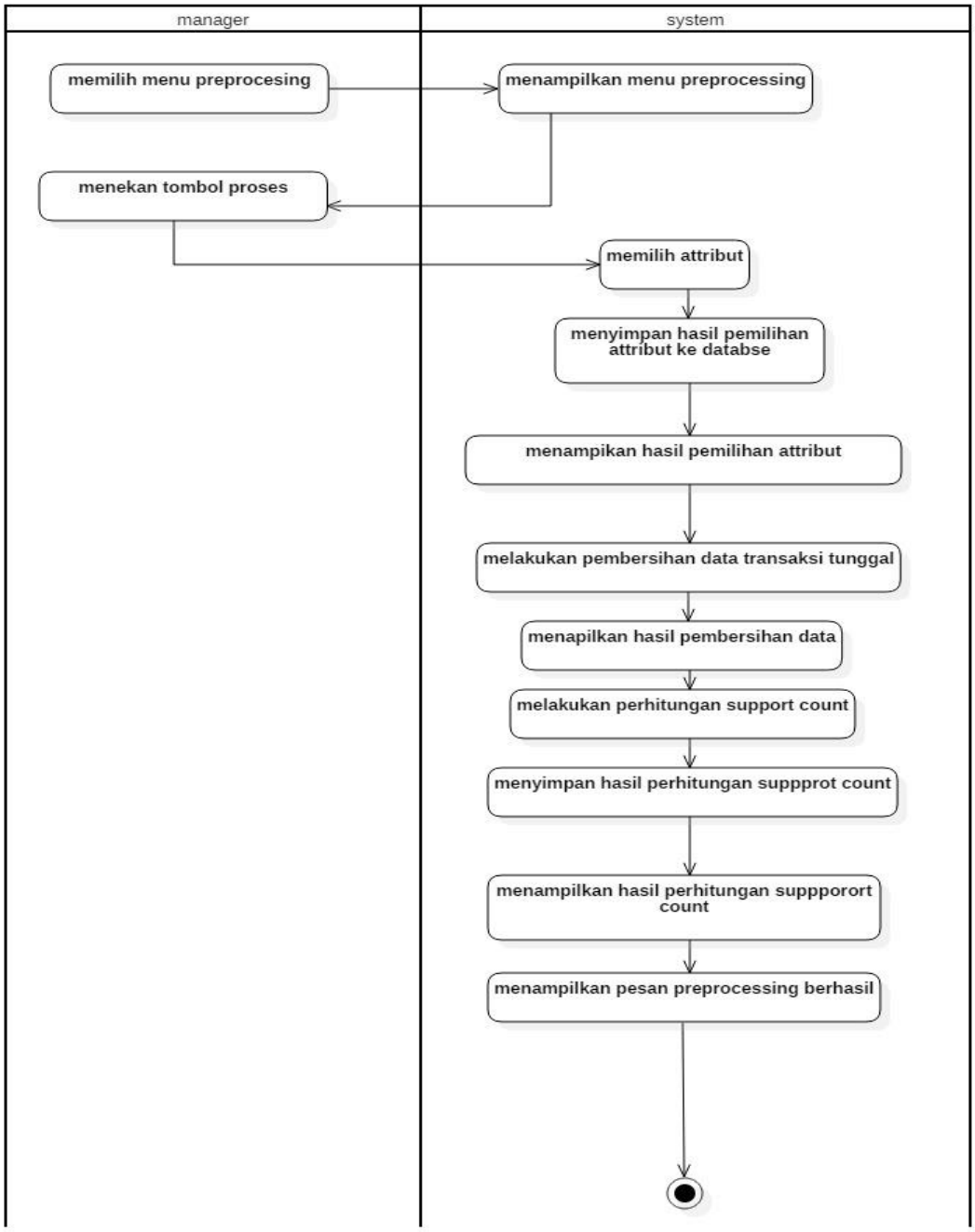
Activity Diagram *Import* menjelaskan aktivitas sistem dalam melakukan proses *Import* data transaksi ke dalam *database*. *Activity* Diagram *Import* dapat dilihat pada gambar 3.50:



Gambar 3. 50 *Activity* Diagram *Import*

4. *Activity* Diagram Preprocesing

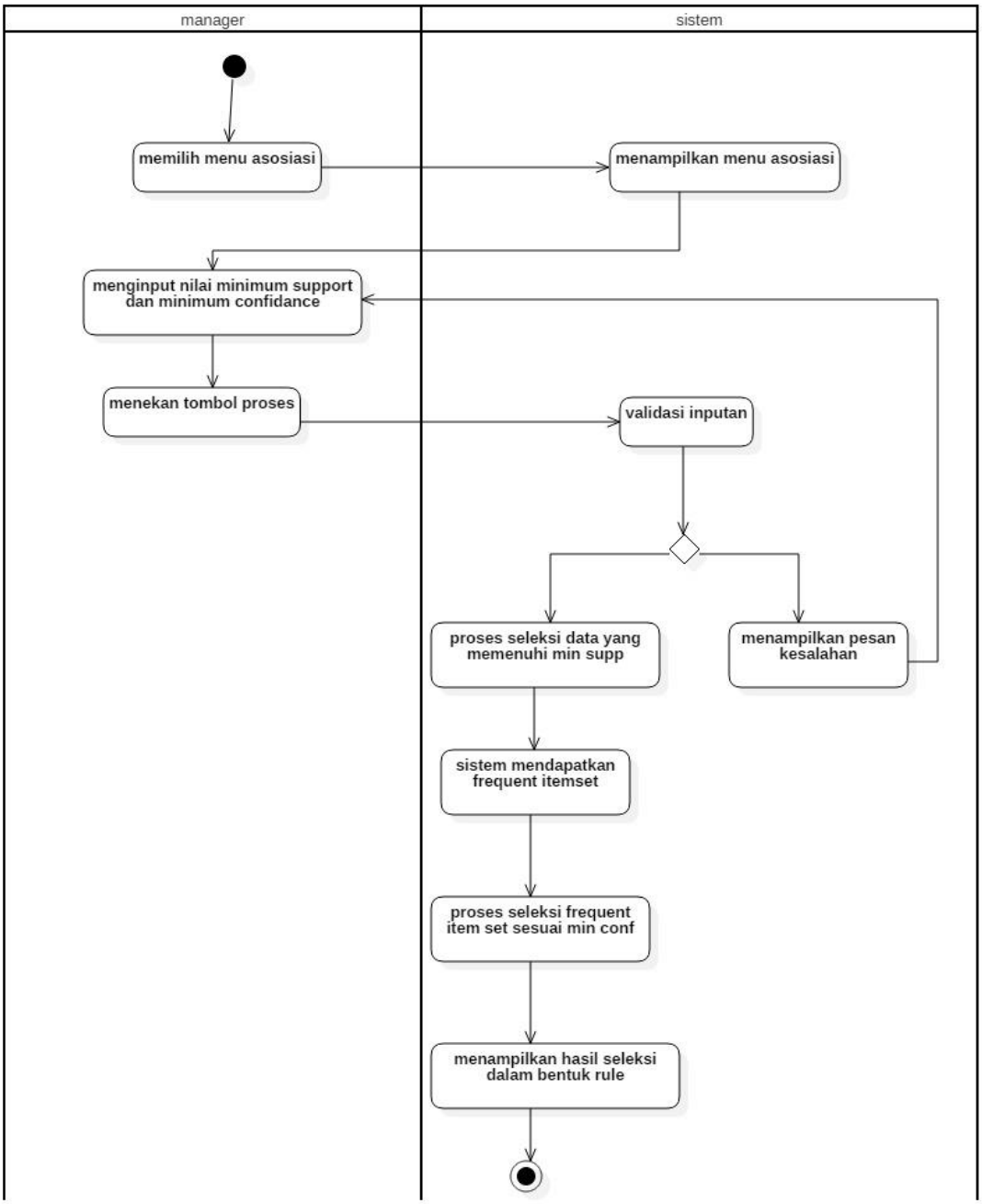
Activity Diagram *Preprocessing* menjelaskan aktivitas sistem dalam melakukan proses Pemilihan Atribut dan Pembersihan Data. *Activity* Diagram *preprocessing* dapat dilihat pada gambar 3.51:



Gambar 3. 51 *Activity* Diagram *Preprocessing*

5. *Activity* Diagram Asosiasi

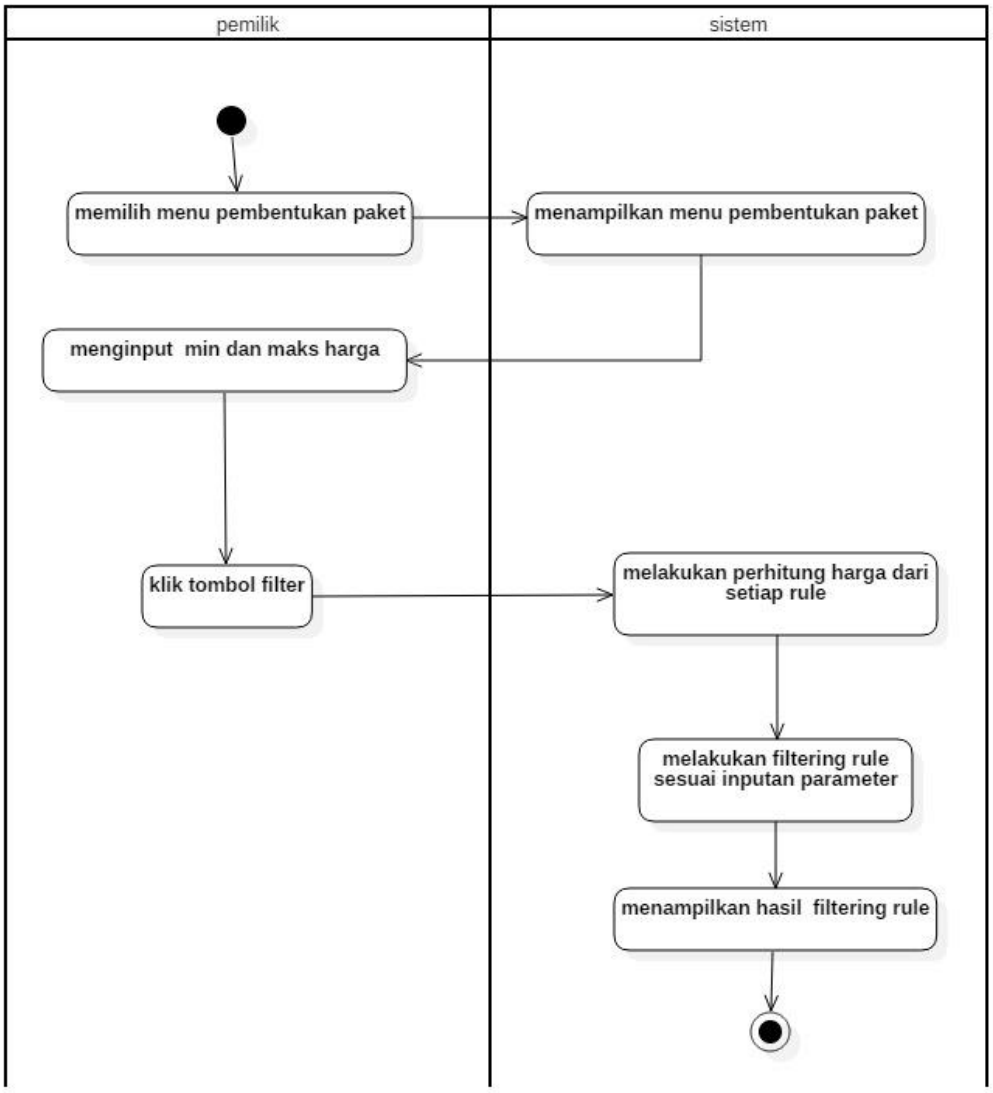
Activity Diagram asosiasi menjelaskan aktivitas user dan sistem dalam melakukan proses input parameter dan melihat *rule*. *Activity* Diagram Asosiasi dapat dilihat pada gambar 3.52:



Gambar 3. 52 *Activity* Diagram Asosiasi

8. *Activity Diagram* Pembentukan Paket

Activity diagram pembentukan paket menjelaskan aktivitas *user* dan sistem dalam menentukan minimal harga dan maksimal harga sesuai dengan yang dimasukkan oleh user. *Activity diagram* lihat hasil pembentukan paket dapat dilihat pada gambar 3.53:



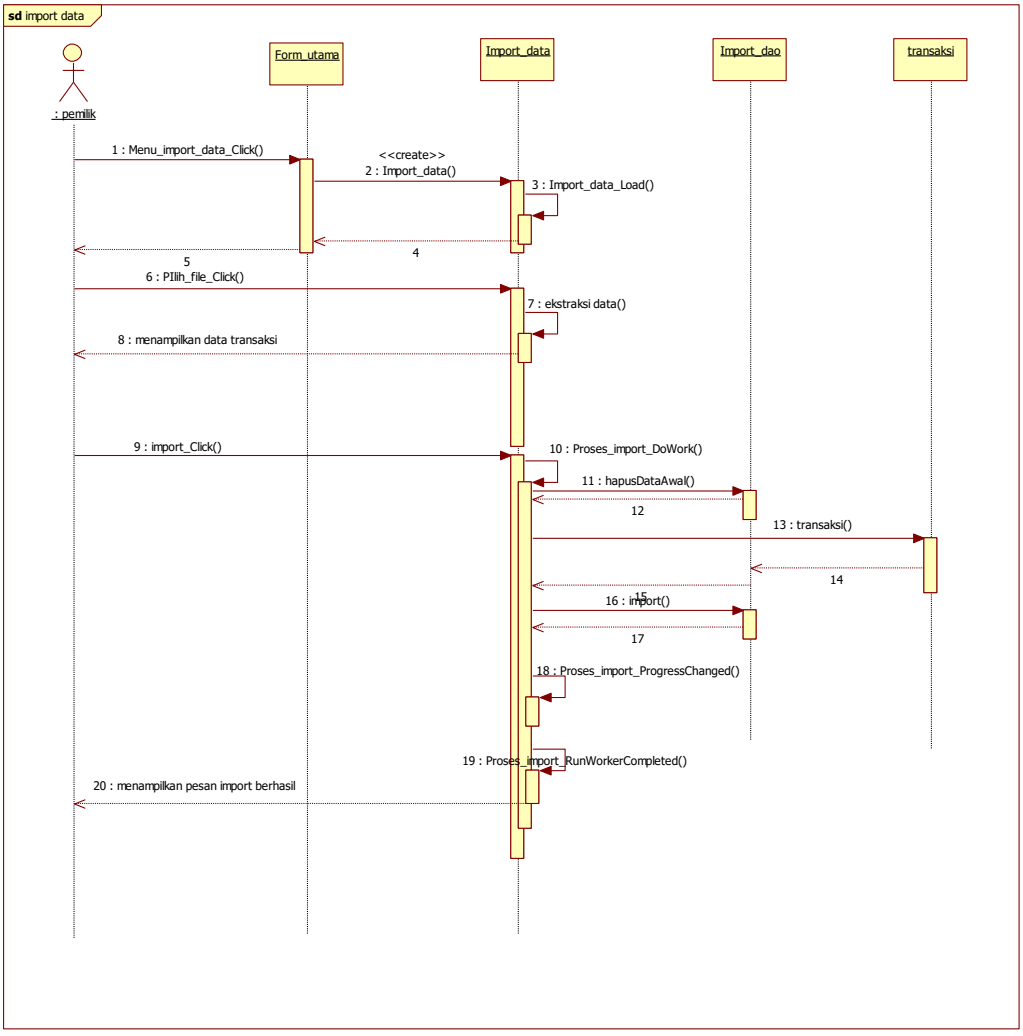
Gambar 3. 53 *Activity* Diagram pembentukan paket

3.1.4.3. 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 *Import* Data

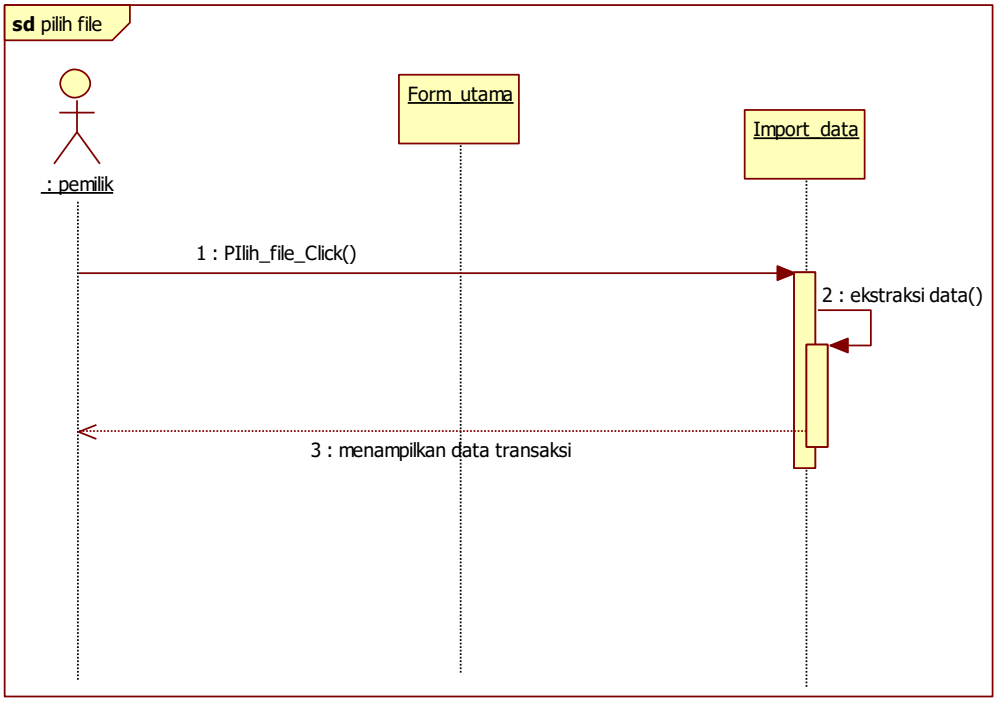
Sequence Diagram pada proses *import* data dapat dilihat pada gambar 3.54.



Gambar 3. 54 *Sequence* Diagram *Import* Data

2. *Sequence* pilih *File*

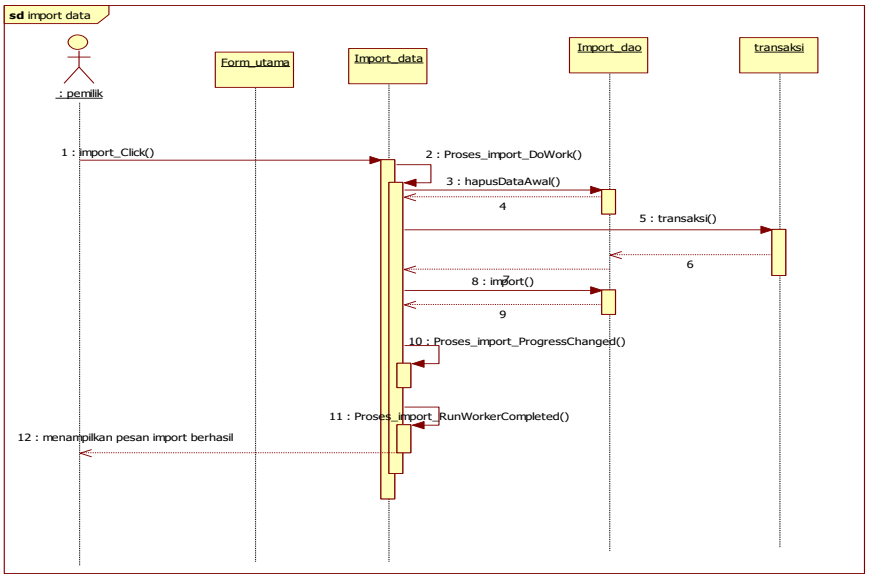
Sequence Diagram pada proses pilih *File* dapat dilihat pada gambar 3.55.



Gambar 3. 55 *Sequence* Diagram Pilih *File*

3. *Sequence import*

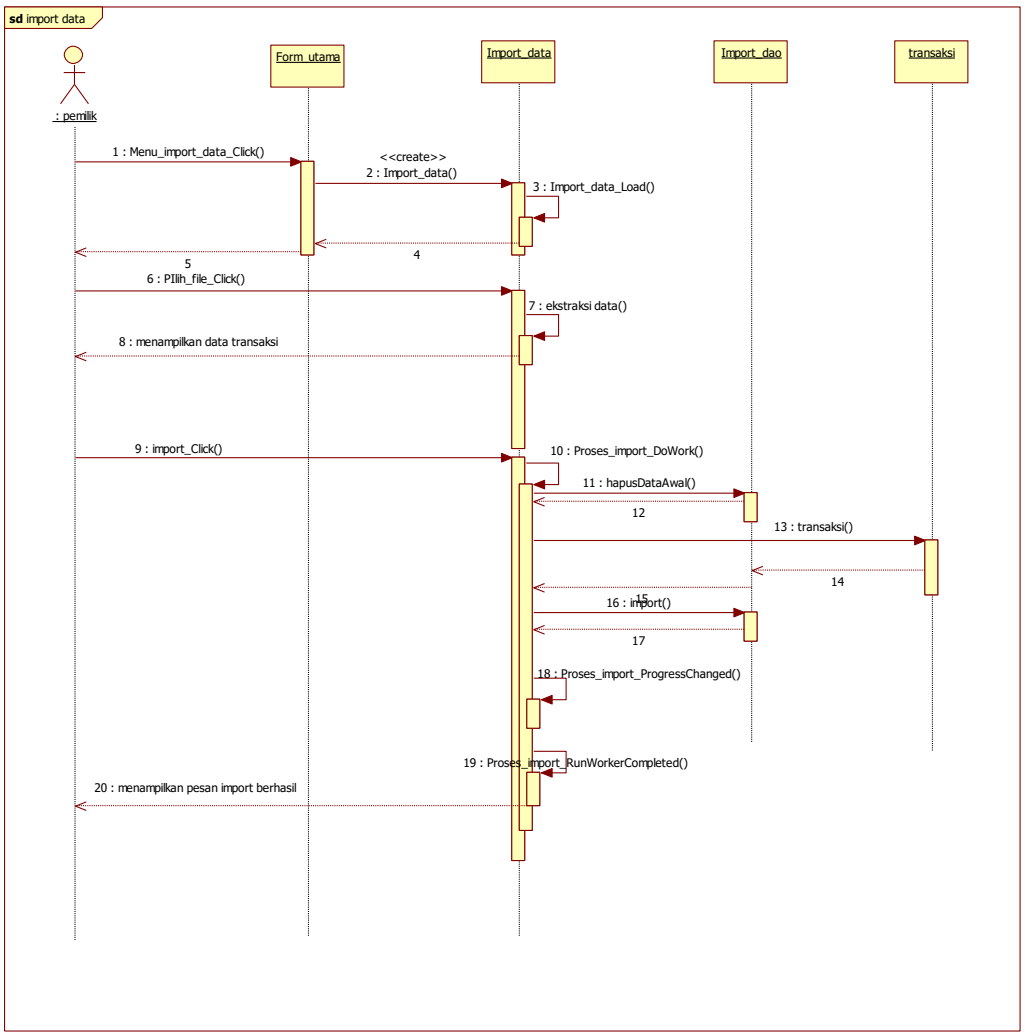
Sequence Diagram pada proses *import* dapat dilihat pada gambar 3.56.



Gambar 3. 56 *Sequence* Diagram *import*

4. *Sequence* Diagram *Preprocessing*

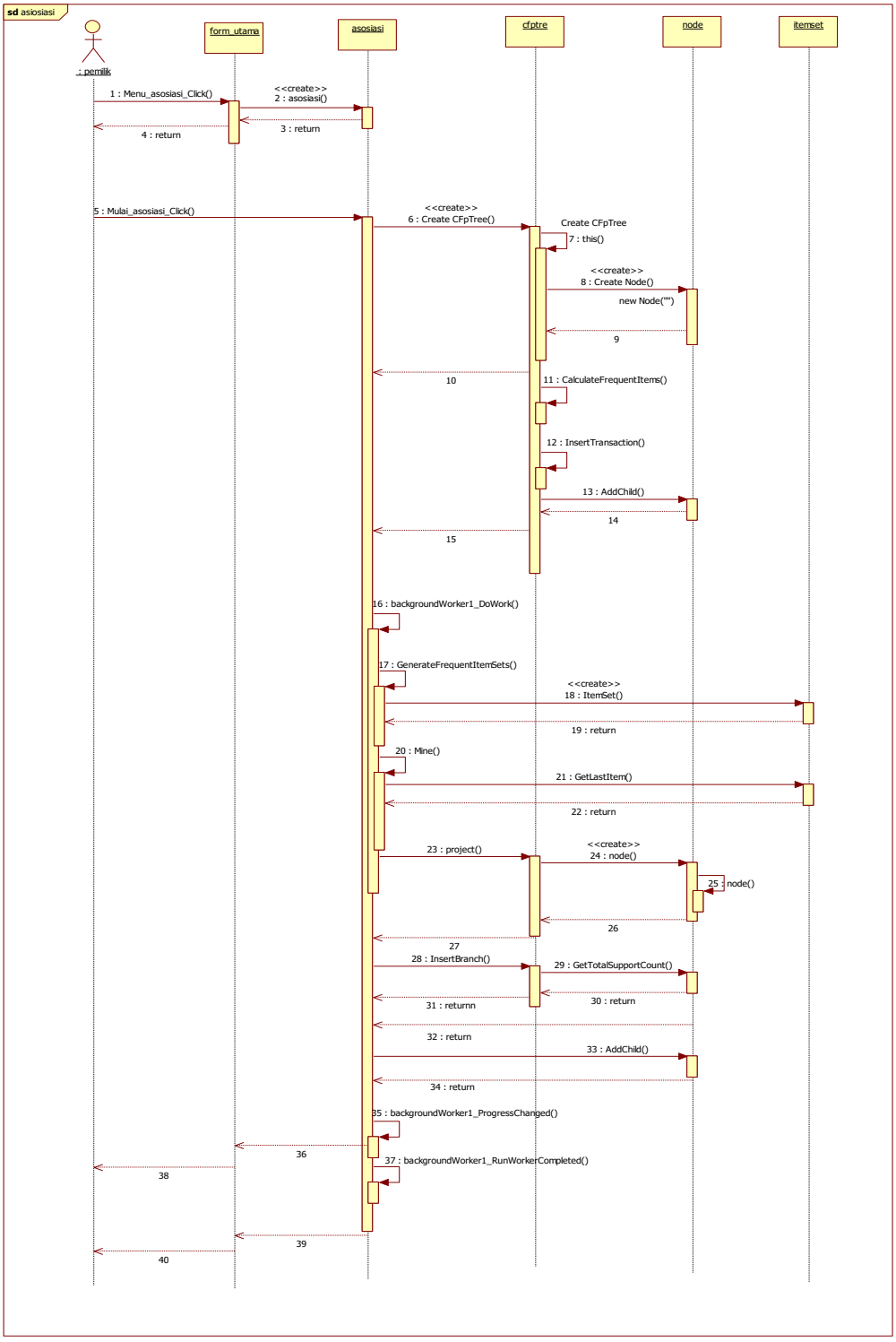
Sequence Diagram *Preprocessing* dapat dilihat pada gambar 3.57



Gambar 3. 57 *Sequence* Diagram *prepossessing*

5. *Sequence* Diagram Proses Asosiasi

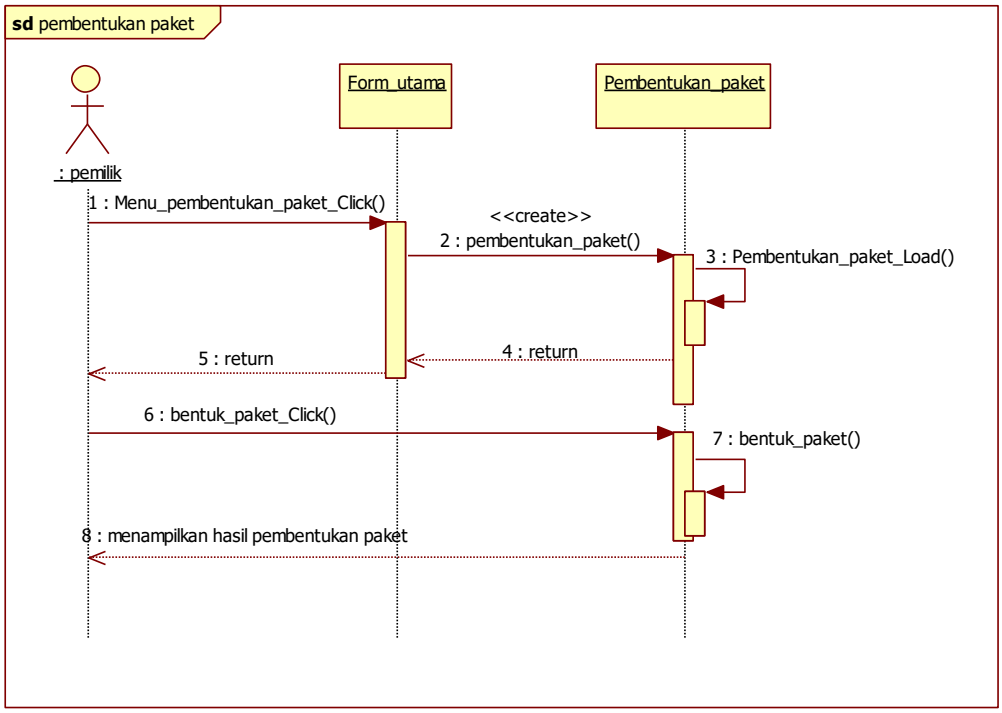
Sequence Diagram pada proses asosiasi dapat dilihat pada gambar 3.58



Gambar 3. 58 *Sequence* Diagram Proses Asosiasi

6. *Squence Diagram* Proses Pembentukan Paket

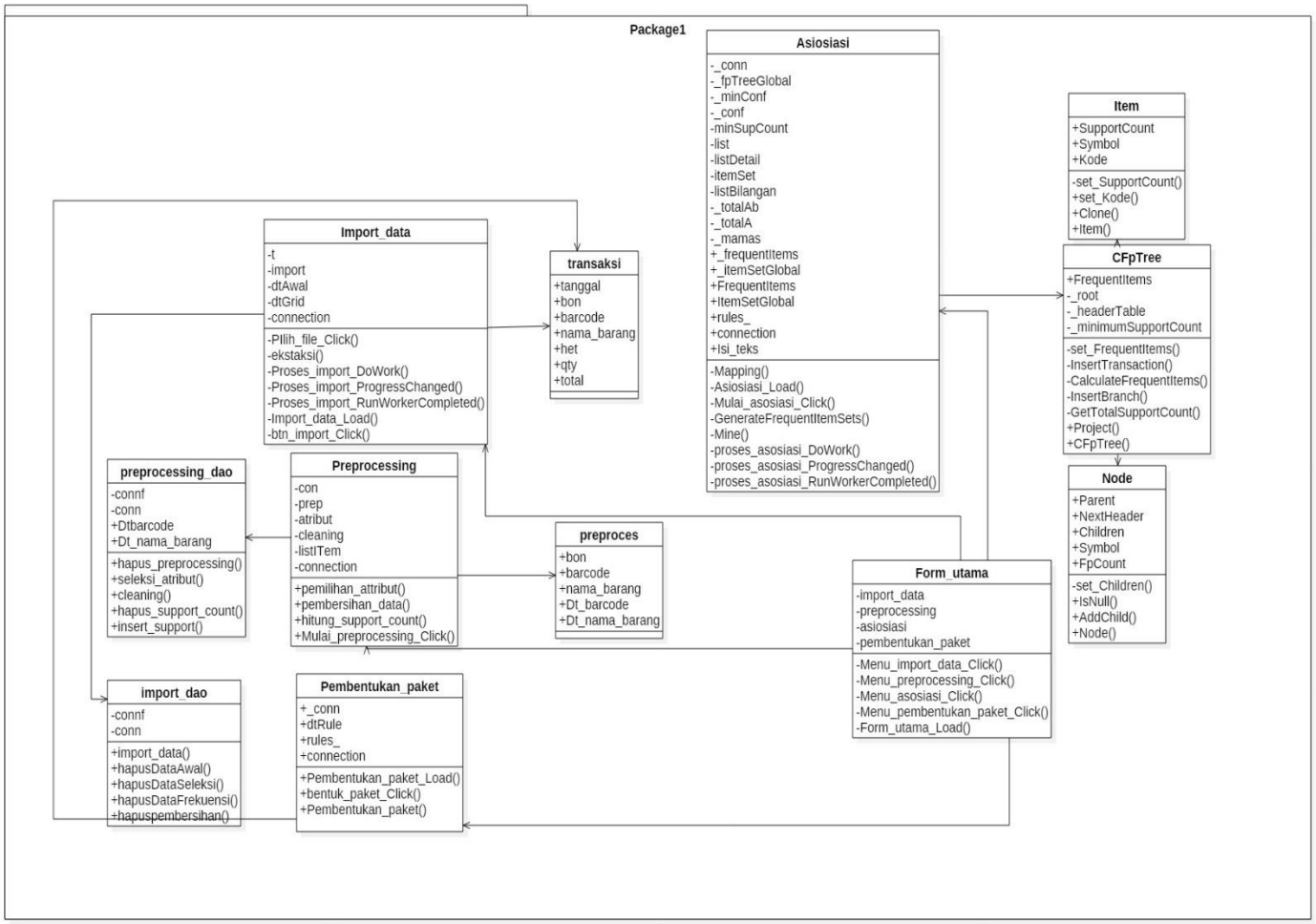
   Sequence diagram pada Proses Pembentukan Paket dapat dilihat pada gambar 3.59



Gambar 3. 59 *Sequence* Diagram Proses Lihat *Rule*s

3.1.4.4 Diagram Class

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 3.60



Gambar 3. 60 *Class Diagram*

3.1 Perancangan Sistem

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

3.2.1 Perancangan Kelas

Berikut adalah perancangan *Class Diagram* yang digunakan:

Tabel 3. 61 Struktur Kelas Form_Utama

Class: Form_Utama		
Nama	Jenis	Type
*import*_data	Atribut	Private
preprocessing	Atribut	Private
asiosiasi	Atribut	Private
Nama	Jenis	Type
pembentukan_paket	Atribut	Private
Menu_Import_data_Click()	Method	Private
Menu_preprocessing_Click()	Method	Private
Menu_asiosiasi_Click()	Method	Private
Menu_pembentukan_paket_Click()	Method	Private
Form_utama_Load()	Method	Private

Tabel 3. 62 Struktur Kelas *Import*_data

Class: *Import*_data		
Nama	Jenis	Type
t	Atribut	private
Import	Atribut	Private
dtAwal	Atribut	Private
dtGrid	Atribut	Private
Connection	Atribut	Private
Plih_file_Click()	Method	Private
ekstaksi()	Method	Private
Proses_import_DoWork()	Method	Private
Proses_import _ProgressChanged()	Method	Private
Proses_import _RunWorkerCompleted()	Method	Private
*Import*_data_Load()	Method	Private
Import_data_Click()	Method	Private

Tabel 3. 63 Struktur Kelas *Import*_dao

Class: *Import*_dao		
Nama	Jenis	Type
Connf	Atribut	Private
Conn	Atribut	Private
*import*_data()	Method	Public
hapusDataAwal()	Method	Public
hapusDataSeleksi()	Method	Public
Nama	Jenis	Type
hapusDataFrekuensi()	Method	Public
hapuspembersihan()	Method	Public

Tabel 3. 64 Struktur Kelas *preprocessing*

Class: *preprocessing*		
Nama	Jenis	Type
Con	Atribut	Public
Prep	Atribut	Public
Atribut	Atribut	Public
Cleaning	Atribut	Public
listITem	Atribut	Public
Connection	Atribut	Public
pemilihan_atribut()	Method	Private
pembersihan_data()	Method	Private
hitung_*support*_count()	Method	Private
proses_Click()	Method	Private

Tabel 3. 65 Struktur Kelas *preprocessing*_dao

Class: *preprocessing*_dao		
Nama	Jenis	Type
Connf	Atribut	Private
Conn	Atribut	Prvate
Dtbarcode	Atribut	Public
hapus_*preprocessing*()	Method	Public
seleksi_atribut()	Method	Public
cleaning()	Method	Public
hapus_*support*_count()	Method	Public
insert_*support*()	Method	Public

Tabel 3. 66 Struktur Kelas preproces

Class: preproces		
Nama	Jenis	Type
Bon	Atribut	Public
Barcode	Atribut	Public
nama_barang	Atribut	Public
Dt_barcode	Atribut	Public
Dt_nama_barang	Atribut	Public

Tabel 3. 67 Struktur Kelas Asosiasi

Class: Asosiasi		
Nama	Jenis	Type
_conn	Atribut	Private
_fpTreeGlobal	Atribut	Private
_minConf	Atribut	Private
_conf	Atribut	Private
minSupCount	Atribut	Private

Nama	Jenis	Type
List	Atribut	Private
listDetail	Atribut	Private
itemSet	Atribut	Private
listBilangan	Atribut	Private
_totalAb	Atribut	Private
_totalA	Atribut	Private
_mamas	Atribut	Private
_*frequent*Items	Atribut	Public
_itemSetGlobal	Atribut	Public
*Frequent*Items	Atribut	Public
ItemSetGlobal	Atribut	Public
*rule*s_	Atribut	Public
Connection	Atribut	Public
Isi_teks	Atribut	Public
Mapping()	Method	Private
Asiosiasi_Load()	Method	Private
btnProses_Click()	Method	Private
Generate*Frequent*ItemSets()	Method	Private
Mine()	Method	Private
Proses_asiosiasi_DoWork()	Method	Private
Proses_asiosiasi _ProgressChanged()	Method	Private
Proses_asiosiasi _RunWorkerCompleted()	Method	Private

Tabel 3. 68 Struktur Kelas Pembentukan_paket

Class: form_paket		
Nama	Jenis	Type
_conn	Atribut	Public
dtRule	Atribut	Public
*rule*s_	Atribut	Public
Connection	Atribut	Public
Pembentukan_paket_Load()	Method	Public

Nama	Jenis	Type
Bentuk_paket_Click()	Method	Public
Pembentukan_paket()	Method	Public

Tabel 3. 69 Struktur Kelas CFpTree

Class: CfpTree		
Nama	Jenis	Type
*Frequent*Items	Atribut	Public
_root	Atribut	Private
_headerTable	Atribut	Private
*_minimumSupport*Count	Atribut	Private
set_*Frequent*Items()	Method	Private
InsertTrans*Action*()	Method	Private
Calculate*Frequent*Items()	Method	Private
InsertBranch()	Method	Private
GetTotal*Support*Count()	Method	Private
Project()	Method	Public
CFpTree()	Method	Public

Tabel 3. 70 Struktur Kelas *Node*

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

Nama	Jenis	Type
Node()	Method	Public

Tabel 3. 71 Struktur Kelas Item

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

Tabel 3. 72 Struktur Kelas Itemset

Class: Itemset		
Nama	Jenis	Type
Items	Atribut	Public
SupportCount	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

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 Frekuensi
4. Data *Rule*s

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 3.15 sampai tabel 3.20.

1. Tabel transaksi

   Tabel transaksi merupakan tabel yang digunakan untuk menyimpan data transaksi hasil *import* dari *File* excel dapat dilihat pada tabel 3.73 di bawah:

Tabel 3. 73 Data Transaksi

Nama Field	Tipe Data	Panjang	keterangan
Id	Int	11	Primari Key
tanggal	Varchar	20	
Bon	Varchar	20	
Barcode	Varchar	30	
Nama_barang	Varchar	30	
Het	float		
Qty	Int		
Total	float	11	

2. Tabel *preprocessing*

Tabel selection adalah tabel yang digunakan untuk menyimpan data hasil *selection data* dapat dilihat pada tabel 3.74 di bawah:

Tabel 3. 74 Data *Preprocessing*

Nama Field	Tipe Data	Panjang	Keterangan
Bon	varchar	30	
Barcode	varchar	30	
Nama_barang	varchar	30	
Id	int	11	*Foreign Key* dari tabel transaksi

3. Tabel frekuensi

Tabel frekuensi adalah tabel yang digunakan untuk menyimpan data hasil penghitungan kemunculan banyaknya masing-masing jenis barang dapat dilihat pada tabel 3.75 di bawah:

Tabel 3. 75 Data Frekuensi

Nama Field	Tipe Data	Panjang	keterangan
Barcode	varchar	30	
frekuensi	int	11	
Id	int	11	*Foreign Key* dari tabel transaksi

4. Tabel *rule*s

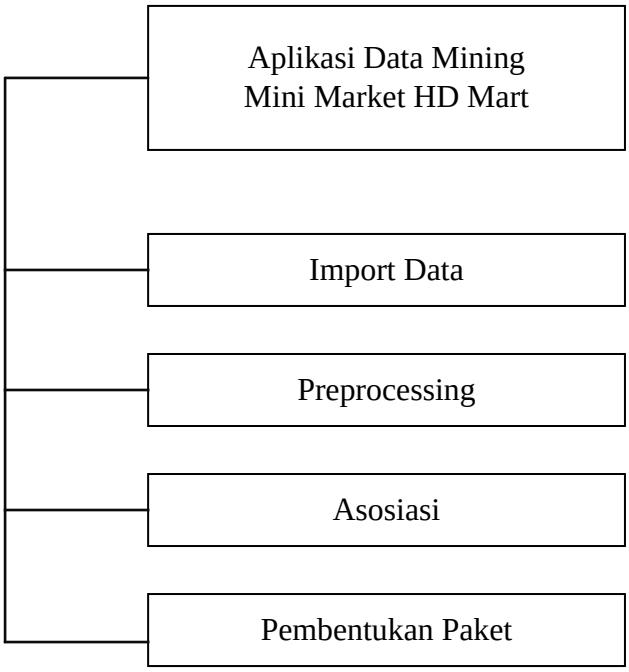
Tabel *rule*s adalah tabel yang digunakan untuk menyimpan hasil pencarian *Frequent Itemset* dapat dilihat pada tabel 3.76 di bawah:

Tabel 3. 76 Data Hasil Pembentukan *Rule*s

Nama Field	Tipe Data	Panjang	keterangan
Rule	text		
Support	int	11	
Confidence	float		
Id	int	11	*Foreign Key* dari tabel transaksi

3.1.2 Perancangan Struktur 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 3.61.



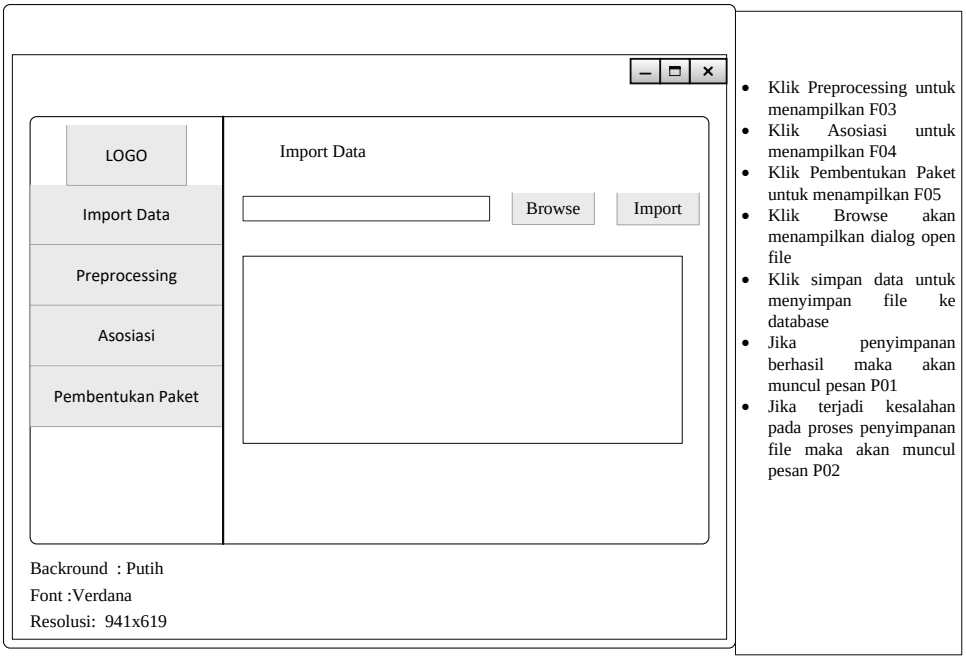
Gambar 3. 61 Struktur Menu

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* HD Mart adalah sebagai berikut:

1. Perancangan Antar Muka *Import Data*

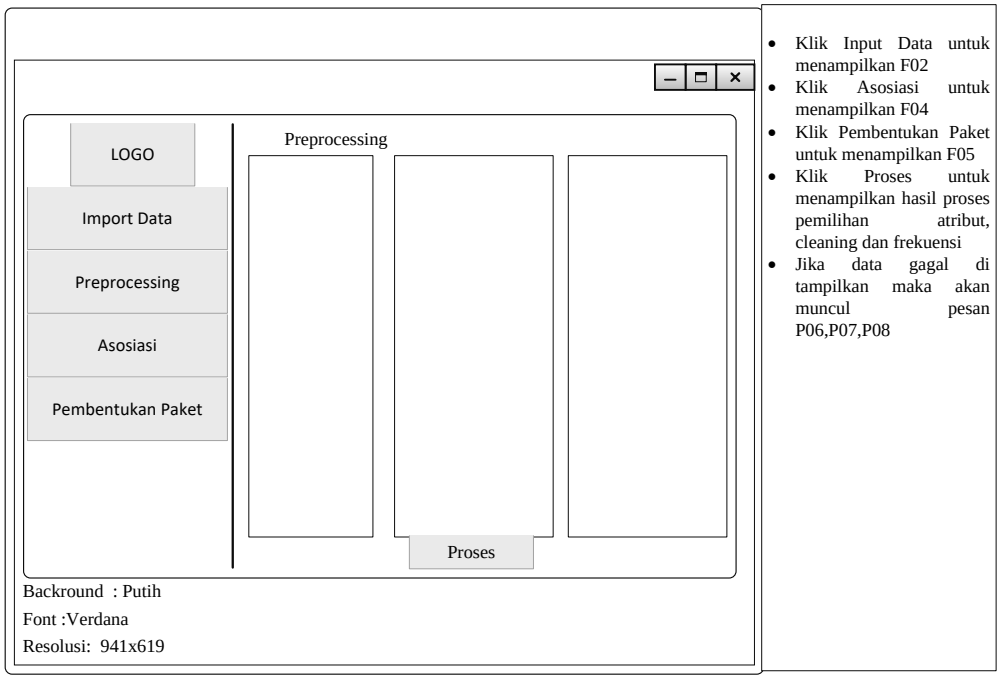
   Perancangan Antar Muka *Import Data* dapat dilihat pada gambar 3.62.



Gambar 3. 62 Antar Muka *Import* Data

2. Perancangan Antar Muka *Preprocessing*

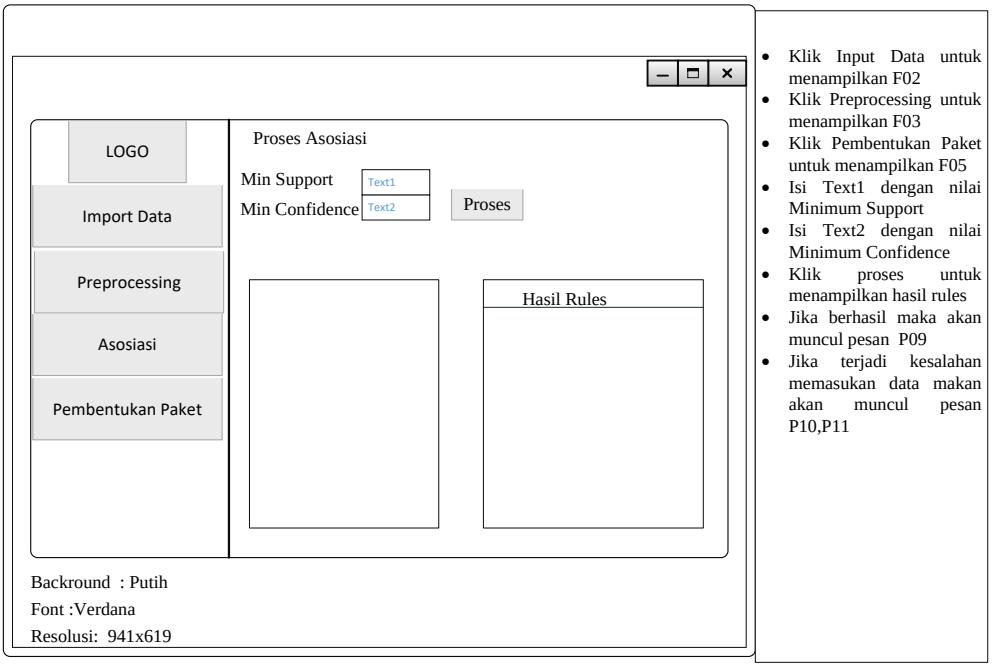
   Perancangan Antar Muka *Preprocessing* dapat dilihat pada gambar 3.63



Gambar 3. 63 Antar Muka *Preprocessing* Data

3. Perancangan Antar Muka Proses Asosiasi

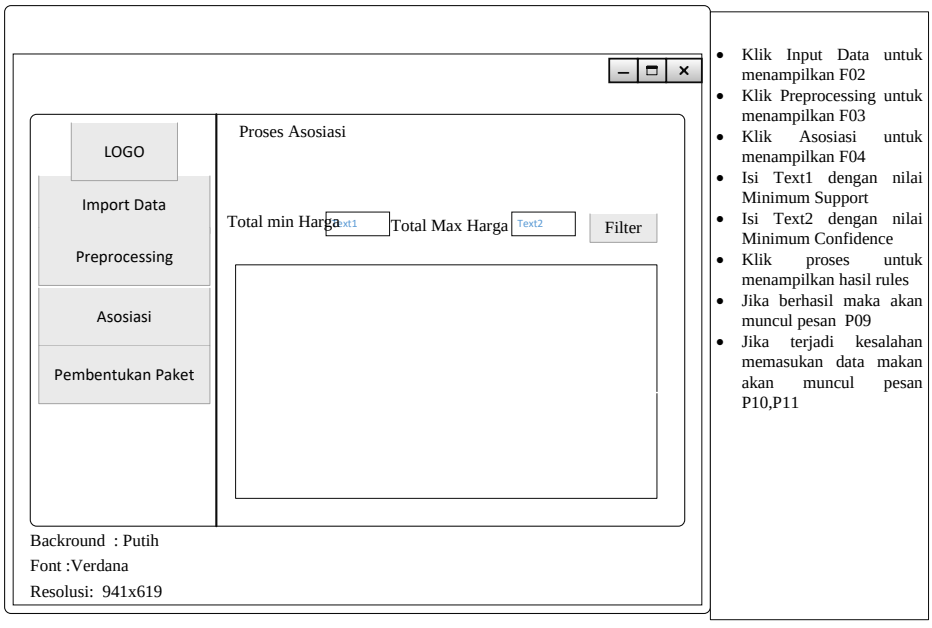
Perancangan Antar Muka Proses Asosiasi dapat dilihat pada gambar 3.64



Gambar 3. 64 Antar Muka Proses Asosiasi

4. Perancangan Antar Muka Proses Pembentukan Paket

Perancangan Antar Muka Proses Pembentukan Paket dapat dilihat pada gambar 3.65



Gambar 3. 65 Antar Muka Proses Pembentukan Paket

3.2.5 Perancangan Pesan

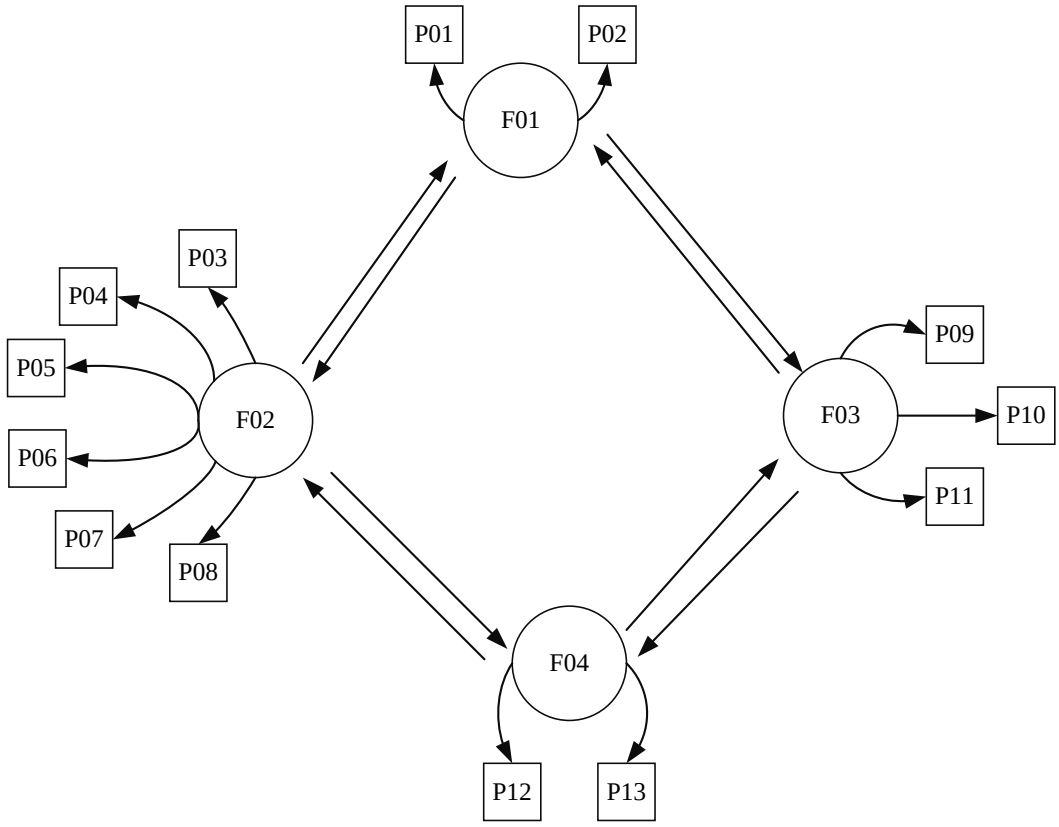
Adapun perancangan pesan dapat dilihat pada gambar 3.66 berikut ini.



Gambar 3. 66 Perancangan Pesan

3.2.6 Perancangan Semantik

Perancangan Semantik dapat dilihat pada gambar 3.67



Gambar 3. 67 Perancangan Semantik