BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

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

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

3.1.1 Analisis Masalah

Berdasarkan hasil pengamatan, diketahui bahwa permasalahan yang ada di rumah makan Laksana Hati yaitu dalam menetukan paket nasi kotak yang tidak sesuai dengan keinginan konsumen karena dalam pemesanan paket nasi kotak rata-rata konsumen memesan tidak berdasarkan paket yang sudah ada melainkan mengkombinasikan paket nasi kotaknya secara sendiri, hal tersebut membuat pihak rumah makan harus menghitung ulang paket nasi kotak yang dipesan oleh konsumen.

3.1.2 Analisis Crisp-DM

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

3.1.2.1 Business *Understanding*

Pemahaman Bisnis atau disebut dengan *Business Understanding* merupakan tahapan pertama yang dilakukan dalam kerangka kerja CRISP-DM. Dalam tahapan bisnis ini terbagi menjadi beberapa bagian, yaitu:

1. Tujuan Bisnis

Tujuan bisnis rumah makan Laksana Hati yaitu memasarkan menu makanan secara langsung untuk memenuhi permintaan konsumen.

2. Menilai Situasi

Menilai situasi merupakan tahapan untuk menjelaskan sumber daya yang tersedia di rumah makan Laksana Hati. Berikut merupakan sumber daya yang ada pada rumah makan Laksna hati.

a. *Inventory of resource*

1. Sumber daya perangkat keras
   a) Terdapat alat penunjang transaksi yaitu printer thermal dan laci kasir
   b) Komputer
2. Sumber daya data
   a) Sumber daya data yang ada di rumah makan Laksana Hati meliputi data transaksi penjualan berupa data *Microsoft Excel* dan bon pemesanan.
3. Sumber daya personil
   a) Pemilik membuat perencanaan, strategi dan kebijakan yang menyangkut operasi rumah makan Laksana Hati.
   b) Juru masak memasak semua menu yang di pesan pelanggan
   c) Cetring minuman menyiapkan minuman yang di pesan pelanggan
   d) Karyawan membantu dalam melayanai pesanan-pesanan pelanggan
   e) Adm.Keuangan melayani pelanggan dalam transaksi pembayarann
   f) House keeping Bertanggung jawab terhadap kebersihan, baik di luar ataupun di dalam lingkungan rumah makan.

b. *Requirements, assumption and constraints*

1. Kebutuhan
   a. Data transaksi yang digunakan dalam penelitian ini adalah data transaksi periode bulan Desember 2015

1 Asumsi

a. Data transaksi yang digunakan merupakan hasil export dari aplikasi di rumah makan Laksana Hati dengan format *Microsoft Excel.*

3. Batasan

a. Hasil dari *data mining* berupa *rule* sebagai acuan untuk membuat paket nasi kotak

b. Hasil *generate rule* akan dilakukan *filtering rule*, yaitu memilih beberapa menu yang ada di dalam *rule* dan menjumlahkan harga dari menu yang sudah dipilih.

c. Dalam satu paket nasi kotak minimal isi tiga meu

d. Batasan untuk pembentukan paket berupa harga minimal dan maksimal paket akan ditentukan oleh pengguna.

3. Penentuan Sasaran Data *Mining*

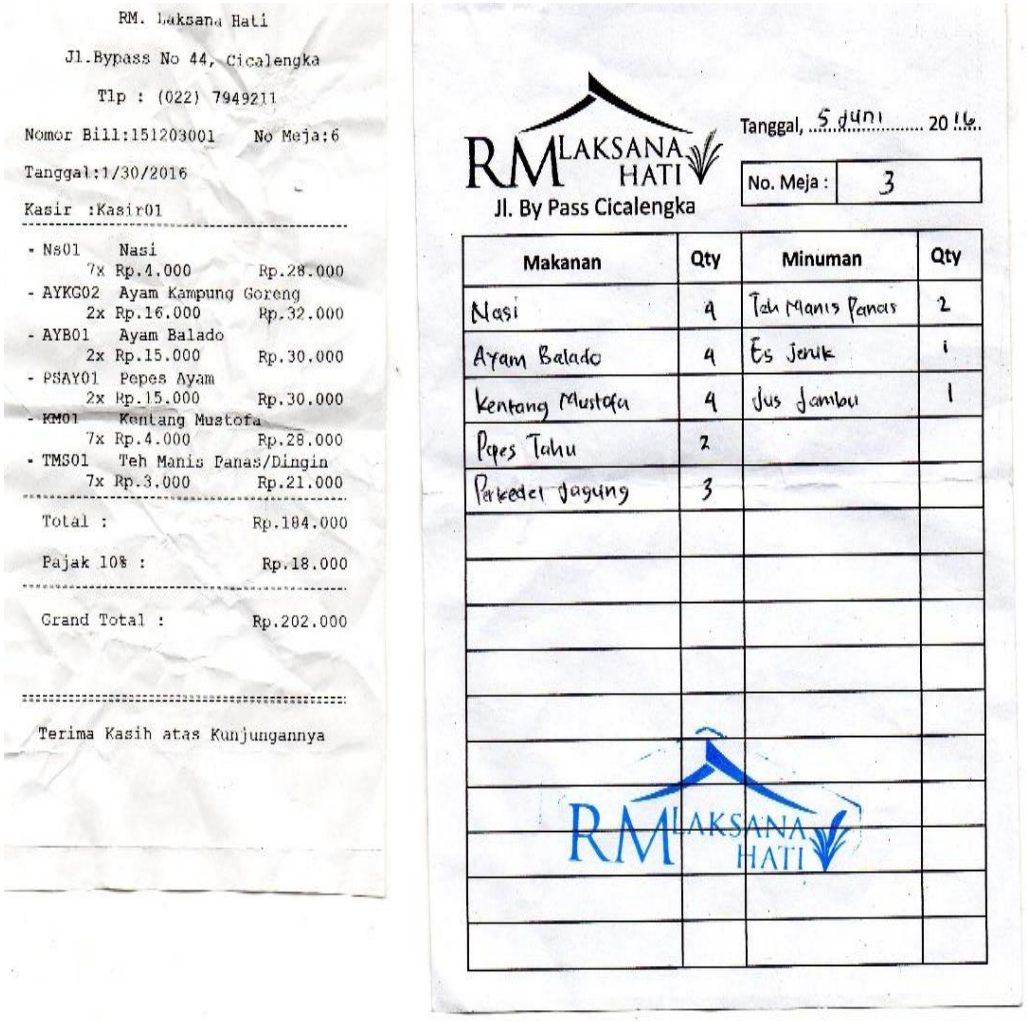
Tujuan dari penerapan Data *Mining* ini adalah untuk mengetahui pasangan menu makanan yang sering dibeli oleh konsumen atau informasi mengenai pola pembelian konsumen yang sering terjadi yang akan dijadikan dasar dalam penentuan pembuatan paket nasi kotaknya.

3.1.2.2 Data *Understanding*

Data Understanding merupakan tahapan kedua yang dilakukan dalam kerangka kerja CRISP-DM. Dalam tahapan data *understanding* ini terbagi menjadi beberapa bagian, yaitu:

a. Mengumpulkan data awal

Sumber data yang didapat dalam penelitian ini merupakan data transaksi penjualan yang terjadi di Rumah Makan Laksana Hati. Dapat dilihat pada gambar 3.1



RM. Laksana Hati
Jl.Bypass No 44, Cicalengka
Tlp : (022) 7949211

Nomor Bill:151203001 No Meja:6
Tanggal:1/30/2016
Kasir :Kasir01

- NS01 Nasi		
7x Rp.4.000		Rp.28.000
- AYKG02 Ayam Kampung Goreng		
2x Rp.16.000		Rp.32.000
- AYB01 Ayam Balado		
2x Rp.15.000		Rp.30.000
- PSAY01 Pepes Ayam		
2x Rp.15.000		Rp.30.000
- KM01 Kentang Mustofa		
7x Rp.4.000		Rp.28.000
- TMS01 Teh Manis Panas/Dingin		
7x Rp.3.000		Rp.21.000
Total :		Rp.184.000
Pajak 10% :		Rp.18.000
Grand Total :		Rp.202.000

Terima Kasih atas Kunjungannya

RM LAKSANA HATI
Jl. By Pass Cicalengka

Tanggal, 5 Juni 2016
No. Meja : 3

Makanan	Qty	Minuman	Qty
Nasi	4	Teh Manis Panas	2
Ayam Balado	4	Es Jeruk	1
Kentang Mustofa	4	Jus Jambu	1
Pepes Tahu	2		
Paketan Jagung	3		

RM LAKSANA HATI

Gambar 3. 1 Data Transaksi Penjualan

b. Mendeskripsikan data

Tahap kedua dalam pemahaman data adalah mendeskripsikan data dengan tujuan memahami data yang didapatkan dari hasil pengumpulan data awal. Adapun detail informasi mengenai data transaksi penjualan yang digunakan dapat dilihat pada table 3.1 dibawah ini.

Tabel 3. 1 Struktur Data Transaksi Penjualan

Dokumen	Keterangan		
Detail Data Transaksi	Deskripsi	Data ini berisi mengenai data transaksi dan pemesanan yang ada di rumah makan Laksana Hati.	
	Format	Microsoft Excel (.xlsx)	
	Atrribut	Nomor Bill	Nomor struk
		No Meja	Nomor meja
		Tanggal	Tanggal percetakan struk
		Kasir	Nama kasir
		Kode Menu	Kode pada setiap menu
		Nama Menu	Nama menu yang dibeli
		Kategori	Jenis menu yang dibeli
		Harga	Harga menu yang dibeli
		Qty	Jumlah menu yang dibeli
		Sub Total	Kuantiti dikali harga
		Total	Jumlah harga dari satu transaksi
		Pajak 10%	Pajak 10% dari rumah makan
		Grand Total	Total harga akhir dalam satu transaksi
		Tunai	Tunai pembayaran dari konsumen
		Kembalian	Jumlah kembalian dari pembayaran

c. Mengidentifikasi masalah kualitas data

Setelah proses mendeskripsikan data transaksi pada tabel 3.1, selanjutnya dilakukan mengidentifikasian kualitas data. Dari data transaksi yang ada, belum berkualitas karena terdapat *noise* yaitu terdapat nama menu yang kategori jenis minuman dan data transaksi yang mimiliki item tunggal . Maka data akan di proses lebih lanjut dalam tahap *data preparation.*

3.1.2.3 Data *Preparation*

Persiapan Data merupakan tahap dimana akan dilakukan pemilihan tabel dan *field* yang akan digunakan dalam proses *mining.* Persiapan data dilakukan dengan sebutan *Preprocessing Data. Preprocessing* merupakan hal yang harus dilakukan dalam proses *data mining*, karena tidak semua data atau atribut data dalam data digunakan dalam proses data *mining*. Proses ini dilakukan agar data yang digunakan sesuai dengan kebutuhan. Adapun tahapan-tahapan *preprocessing* data dalam penelitian ini adalah sebagai berikut

1. Pembersihan Data

   Pembersihan Data proses menghilangkan *noise*. Dalam penelitian ini yaitu menghilangkan data transaksi yang memiliki kategori jenis minuman serta data transaksi yang memiliki data tunggal. Setelah proses pembersihan dilakukan yang awalnya 27 transaksi sebanyak 198 record menjadi 27 transaksi sebanyak 157 record. Berikut adalah tabel sebelum pembersihan dan sesudah pembersihan dapat dilihat pada tabel 3.2 dan tabel 3.3

Tabel 3.2 Tabel Sebelum Pembersihan

NomorBill	No Meja	Tanggal	Kasir	Kode Menu	Nama Menu	Kategori	Harga	Qty	Sub Total	Total
151200001	8	12/1/15 12:15	Kasir01	NS01	Nasi	Makanan	Rp4,000	7	Rp28,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	AYB01	Ayam Balado	Makanan	Rp15,000	3	Rp45,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	KM01	Kentang Mustofa	Makanan	Rp4,000	7	Rp28,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	PTA03	Pepes Tahu	Makanan	Rp8,000	2	Rp16,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	PJ01	Perkedel Jagung	Makanan	Rp2,000	2	Rp4,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	TM01	Teh Manis Panas/ Dingin	Minuman	Rp3,000	7	Rp21,000	Rp142,000
151200002	6	12/1/15 12:20	Kasir01	NS01	Nasi	Makanan	Rp4,000	5	Rp20,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	AYB01	Ayam Balado	Makanan	Rp15,000	2	Rp30,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	AYKG02	Ayam Kampung Goreng	Makanan	Rp16,000	3	Rp48,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	PJA02	Pepes Jamur	Makanan	Rp5,500	2	Rp11,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	PTA03	Pepes Tahu	Makanan	Rp8,000	2	Rp16,000	Rp158,000
151200002	6	12/1/1512:20	Kasir01	PJ01	Perkedel Jagung	Makanan	Rp2,000	4	Rp8,000	Rp158,000
151200002	6	12/1/1512:20	Kasir01	JR01	Jeruk Panas/ Dingin	Minuman	Rp8,000	2	Rp16,000	Rp158,000
151200002	6	12/1/1512:20	Kasir01	TM01	Teh Manis Panas/ Dingin	Minuman	Rp3,000	3	Rp9,000	Rp158,000
…	…	……	…..	…..	……	…….	……	…	…….	…….
151200027	8	12/1/1513:05	Kasir01	KM01	Kentang Mustofa	Makanan	Rp4,000	3	Rp12,000	Rp66,000

Tabel 3. 3 Sesudah Pembersihan

NomorBill	No Meja	Tanggal	Kasir	Kode Menu	Nama Menu	Kategori	Harga	Qty	Sub Total	Total
151200001	8	12/1/15 12:15	Kasir01	NS01	Nasi	Makanan	Rp4,000	7	Rp28,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	AYB01	Ayam Balado	Makanan	Rp15,000	3	Rp45,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	KM01	Kentang Mustofa	Makanan	Rp4,000	7	Rp28,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	PTA03	Pepes Tahu	Makanan	Rp8,000	2	Rp16,000	Rp142,000
151200001	8	12/1/15 12:15	Kasir01	PJ01	Perkedel Jagung	Makanan	Rp2,000	2	Rp4,000	Rp142,000
151200002	6	12/1/15 12:20	Kasir01	NS01	Nasi	Makanan	Rp4,000	5	Rp20,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	AYB01	Ayam Balado	Makanan	Rp15,000	2	Rp30,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	AYKG02	Ayam Kampung Goreng	Makanan	Rp16,000	3	Rp48,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	PJA02	Pepes Jamur	Makanan	Rp5,500	2	Rp11,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	PTA03	Pepes Tahu	Makanan	Rp8,000	2	Rp16,000	Rp158,000
151200002	6	12/1/15 12:20	Kasir01	PJ01	Perkedel Jagung	Makanan	Rp2,000	4	Rp8,000	Rp158,000
…	…	……	…..	…..	……	…….	……	…	…….	…….
151200027	8	12/1/1513:05	Kasir01	KM01	Kentang Mustofa	Makanan	Rp4,000	3	Rp12,000	Rp66,000

2. Pemilihan Atribut

Setelah proses pembersihan data kemudian dilakukan proses pemilihan atribut, proses dimana atribut data akan dipilih dan diproses sesuai dengan kebutuhan *data mining*. Dalam penelitian ini, atribut yang dibutuhkan hanya 2 atribut yaitu Nomor Bill dan Nama Menu. Kedua atribut ini digunakan untuk memenuhi tujuan awal dimana akan dicari pola pembelian konsumen berdasarkan menu yang dibeli. Seperti atribut Nomor Bill digunakan untuk membedakan satu transaksi dengan transakaksi lainya, dan atribut Nama Menu digunakan untuk mengetahui menu apa saja yang dibeli dalam satu transaksi. Berikut merupakan tabel pemilihan atribut dapat dilihat pada tabel 3.4

Tabel 3. 4 Pemilihan Atribut

NomorBill	Nama Menu
151200001	Nasi
151200001	Ayam Balado
151200001	Kentang Mustofa
151200001	Pepes Tahu
151200001	Perkedel Jagung
151200002	Nasi
151200002	Ayam Balado
151200002	Ayam Kampung Goreng
151200002	Pepes Jamur
151200002	Pepes Tahu
151200002	Perkedel Jagung
……	……..
151200027	Kentang Mustofa

3.1.2.4 Modeling

Penelitian ini bertujuan untuk mendapatkan informasi mengenai pola pembelian konsumen yang nantinya akan digunakan oleh pihak rumah makan Laksana Hati sebagai dasar pengambilan keputusan untuk menentukan menu apa saja yang bisa dijadikan sebuah paket nasi kotak yang nantinya akan ditawarkan ke konsumen.

Dengan menerapkan *Data Mining* data transaksi akan diolah dengan aturan asosiasi atau metode *Association Rules* untuk menetukan pola-pola pembelian menu yang sering dibeli oleh konsumen. *Association Rule* terbagi menjadi dua tahap, yaitu

1. Mencari *frequent itemset* pada tahap ini menggunakan Algoritma *CT-Pro* dengan langkah sebagai berikut:
   a. Menghitung jumlah kemunculan tiap menu
   b. Seleksi dengan menentukan nilai minimum *support*
   c. Membangun *Global Item Table*
   d. *Mapping* Data
   e. Pembentukan *CFP-Tree*
   f. Penggalian *Frequent Patterns*
2. Mendefinisikan *Condition* dan *Result* (*Generate Rule*)

1. Mencari *Frequent Itemset*

Berikut adalah langkah-langkah dalam mencari *frequent itemset*:

a. Menghitung jumlah kemunculan tiap menu

Masing-masing menu dihitung jumlah kemunculannya. Hasil dari penghitungan kemunculan menu dapat dilihat pada tabel 3.5 dibawah ini.

Tabel 3. 5 Penghitungan Kemunculan Menu

Nama Menu	***Support Count***
Nasi	23
Perkedel Jagung	16
Ayam Kampung Goreng	11
Kentang Mustofa	11
Tempe Orek	10
Ayam Balado	10
Pepes Tahu	10
Ikan Mas Goreng	7
Sop Buntut	7
Udang Goreng	6
Gepuk Sapi	6
Cumi Bumbu	6
Sate Usus	5
Telor Balado	5
Sayur Sop	4
Ati Ampela	4
Pepes Ayam	4
Pepes Ikan Mas	3
Pepes Usus	3

Nama Menu	*Support Count*
Capcay	2
Soto Ayam	2
Pepes Jamur	1
Rolade	1

b. Seleksi dengan menentukan nilai *minimum support*

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

Dalam menentukan nilai *minimum support* dapat berubah sesuai kebutuhan *user.* Hal ini juga diperkuat dengan pernyatan pakar *data mining* Daniel T. Larose, yang menerangkan bahwa *user* bebas menentukan nilai *minimum support* (*minsup*) dan *minimum confidence* (*mincof*) sesuai kebutuhan. Karena, jika nilai *minsup* terlalu rendah maka akan mengakibatkan *rule* asosiasi yang dihasilkan terlalu banyak dan akan ada *rule* yang kurang bermakna. Sebaliknya jika nilai *minimum support* terlalu tinggi maka akan mengakibatkan *rule* asosiasi terlalu sedikit atau bahkan tidak dapat dihasilkan sama sekali.

Maka dalam kasus ini batasan menu yang muncul harus sebanyak >= 5 kali. Berdasarkan tabel 3.5 untuk menu yang *support count*-nya tidak memenuhi batasan *minimum support* dieliminasi. Berikut adalah tabel hasil eliminasi kemunculan menu yang memenuhi *minimun suport* yang diacu pada tabel 3.6.

Tabel 3. 6 Hasil Eliminasi kemunculan Menu

Nama Menu	*Support Count*
Nasi	23
Perkedel Jagung	16
Ayam Kampung Goreng	11
Kentang Mustofa	11
Tempe Orek	10
Ayam Balado	10
Pepes Tahu	10
Ikan Mas Goreng	7
Sop Buntut	7
Udang Goreng	6

Nama Menu	*Support Count*
Gepuk Sapi	6
Cumi Bumbu	6
Sate Usus	5
Telor Balado	5

Untuk menu yang *support count*-nya tidak memenuhi batas *minimum support* juga dielminasi dan hasilnya dapat dilihat pada tabel 3.7

Tabel 3. 7 Data Transaksi yang *frequent*

NomorBill	Nama Menu
151200001	Nasi
151200001	Ayam Balado
151200001	Kentang Mustofa
151200001	Pepes Tahu
151200001	Perkedel Jagung
151200002	Nasi
151200002	Ayam Balado
151200002	Ayam Kampung Goreng
151200002	Pepes Tahu
151200002	Perkedel Jagung
151200003	Nasi
151200003	Ayam Kampung Goreng
151200003	Perkedel Jagung
151200003	Telor Balado
151200003	Udang Goreng
151200004	Nasi
151200004	Cumi Bumbu
151200004	Kentang Mustofa
151200004	Pepes Tahu
151200004	Perkedel Jagung
151200004	Sate Usus
151200004	Tempe Orek
151200005	Nasi
151200005	Ayam Balado
151200005	Ayam Kampung Goreng
151200005	Gepuk Sapi
151200005	Kentang Mustofa
151200005	Pepes Tahu
151200005	Sop Buntut

NomorBill	Nama Menu
151200014	Kentang Mustofa
151200014	Perkedel Jagung
151200014	Telor Balado
151200015	Ikan Mas Goreng
151200015	Perkedel Jagung
151200015	Sop Buntut
151200015	Tempe Orek
151200015	Udang Goreng
151200016	Nasi
151200016	Ayam Kampung Goreng
151200016	Ikan Mas Goreng
151200016	Telor Balado
151200017	Nasi
151200017	Ayam Balado
151200017	Gepuk Sapi
151200017	Kentang Mustofa
151200017	Perkedel Jagung
151200017	Sate Usus
151200017	Sop Buntut
151200017	Udang Goreng
151200018	Nasi
151200018	Ayam Kampung Goreng
151200018	Perkedel Jagung
151200018	Telor Balado
151200019	Ayam Kampung Goreng
151200019	Cumi Bumbu
151200019	Gepuk Sapi
151200019	Tempe Orek
151200020	Nasi

NomorBill	Nama Menu
151200006	Nasi
151200006	Ayam Kampung Goreng
151200006	Sop Buntut
151200006	Tempe Orek
151200007	Ayam Kampung Goreng
151200007	Ikan Mas Goreng
151200007	Perkedel Jagung
151200007	Udang Goreng
151200008	Nasi
151200008	Gepuk Sapi
151200008	Pepes Tahu
151200008	Perkedel Jagung
151200009	Nasi
151200009	Kentang Mustofa
151200009	Pepes Tahu
151200009	Telor Balado
151200009	Tempe Orek
151200010	Nasi
151200010	Ayam Balado
151200010	Ikan Mas Goreng
151200010	Perkedel Jagung
151200010	Udang Goreng
151200011	Nasi
151200011	Ayam Kampung Goreng
151200011	Kentang Mustofa
151200011	Pepes Tahu
151200011	Perkedel Jagung
151200012	Nasi
151200012	Perkedel Jagung
151200012	Tempe Orek
151200013	Nasi
151200013	Ayam Balado
151200013	Ayam Kampung Goreng
151200013	Cumi Bumbu
151200013	Gepuk Sapi
151200013	Pepes Tahu
151200013	Sop Buntut
151200014	Nasi

NomorBill	Nama Menu
151200020	Cumi Bumbu
151200020	Pepes Tahu
151200020	Tempe Orek
151200021	Nasi
151200021	Ayam Balado
151200021	Ayam Kampung Goreng
151200021	Ikan Mas Goreng
151200021	Kentang Mustofa
151200021	Perkedel Jagung
151200021	Sate Usus
151200021	Udang Goreng
151200022	Nasi
151200022	Perkedel Jagung
151200022	Sop Buntut
151200022	Tempe Orek
151200023	Ikan Mas Goreng
151200023	Tempe Orek
151200024	Nasi
151200024	Ayam Balado
151200024	Cumi Bumbu
151200024	Kentang Mustofa
151200024	Pepes Tahu
151200024	Sop Buntut
151200025	Nasi
151200025	Ayam Balado
151200025	Gepuk Sapi
151200025	Perkedel Jagung
151200025	Sate Usus
151200026	Nasi
151200026	Ayam Balado
151200026	Cumi Bumbu
151200026	Kentang Mustofa
151200026	Tempe Orek
151200027	Nasi
151200027	Ikan Mas Goreng
151200027	Kentang Mustofa
151200027	Sate Usus

c. Membangun *Global Item Table*

Berdasarkan tabel 3.7 nama menu yang sudah memenuhi nilai minimum *support*, maka akan dibangun tabel *global item table* dimana masing-masing menu di hitung berdasarkan jumlah kemunculannya. Setelah itu di berikan *id global item* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Pemberian *id global item* ini nantinya akan digunakan pada proses *Mapping* Data. Berikut adalah *global item table* dan *id global item* yang dapat dilihat pada tabel 3.8 dan tabel 3.9.

Tabel 3. 8 *Global Item Table*

Nama Menu	***Support Count***
Nasi	23
Perkedel Jagung	16
Ayam Kampung Goreng	11
Kentang Mustofa	11
Tempe Orek	10
Ayam Balado	10
Pepes Tahu	10
Ikan Mas Goreng	7
Sop Buntut	7
Udang Goreng	6
Gepuk Sapi	6
Cumi Bumbu	6
Sate Usus	5
Telor Balado	5

Tabel 3. 9 *Id Global Item*

Nama Menu	***Support Count***	***Id Global item***
Nasi	23	1
Perkedel Jagung	16	2
Ayam Kampung Goreng	11	3
Kentang Mustofa	11	4
Tempe Orek	10	5
Ayam Balado	10	6
Pepes Tahu	10	7
Ikan Mas Goreng	7	8
Sop Buntut	7	9

Nama Menu	*Support Count*	*Id Global item*
Udang Goreng	6	10
Gepuk Sapi	6	11
Cumi Bumbu	6	12
Sate Usus	5	13
Telor Balado	5	14

d. *Mapping* Data

Mapping yaitu memetakan data transaksi dari tabel 3.7 terhadap *id global item* pada tabel 3.9, dimana nama menu pada tabel 3.7 digantikan oleh *id global item* sesuai dengan tabel 3.11. Data transaksi kemudian diurutkan dari *id global item* terkecil ke terbesar dari setiap Nomor Billnya. Berikut adalah hasil *mapping* data transaksi yang dapat dilihat pada tabel 3.10 di bawah ini.

Tabel 3. 10 Tabel *Mapping*

NomorBill	*Id Global Item*
151200001	1
151200001	2
151200001	4
151200001	6
151200001	7
151200002	1
151200002	2
151200002	3
151200002	6
151200002	7
151200003	1
151200003	2
151200003	3
151200003	10
151200004	1
151200004	2
151200004	4
151200004	5
151200004	7
151200004	12
151200004	13
151200005	1
151200005	3
151200005	4
151200005	6

NomorBill	*Id Global Item*
151200007	3
151200007	8
151200007	10
151200008	1
151200008	2
151200008	7
151200008	11
151200009	1
151200009	4
151200009	5
151200009	7
151200009	14
151200010	1
151200010	2
151200010	8
151200010	10
151200011	1
151200011	2
151200011	3
151200011	4
151200011	7
151200012	1
151200012	2
151200012	5
151200013	1

NomorBill	*Id Global Item*
151200014	4
151200014	14
151200015	2
151200015	5
151200015	8
151200015	9
151200015	10
151200016	1
151200016	3
151200016	8
151200016	14
151200017	1
151200017	2
151200017	4
151200017	9
151200017	10
151200017	11
151200017	13
151200018	1
151200018	2
151200018	3
151200018	14
151200019	3
151200019	5
151200019	11

NomorBill	*Id Global Item*
151200021	4
151200021	6
151200021	8
151200021	10
151200021	13
151200022	1
151200022	2
151200022	5
151200022	9
151200023	5
151200023	8
151200024	1
151200024	4
151200024	6
151200024	9
151200024	12
151200025	1
151200025	2
151200025	6
151200025	11
151200025	13
151200026	1
151200026	4
151200026	5
151200026	6

NomorBill	*Id Global Item*
151200005	7
151200005	9
151200005	11
151200006	1
151200006	3
151200006	5
151200006	9
151200007	2

NomorBill	*Id Global Item*
151200013	3
151200013	6
151200013	7
151200013	9
151200013	11
151200013	12
151200014	1
151200014	2

NomorBill	*Id Global Item*
151200019	12
151200020	1
151200020	5
151200020	7
151200020	12
151200021	1
151200021	2
151200021	3

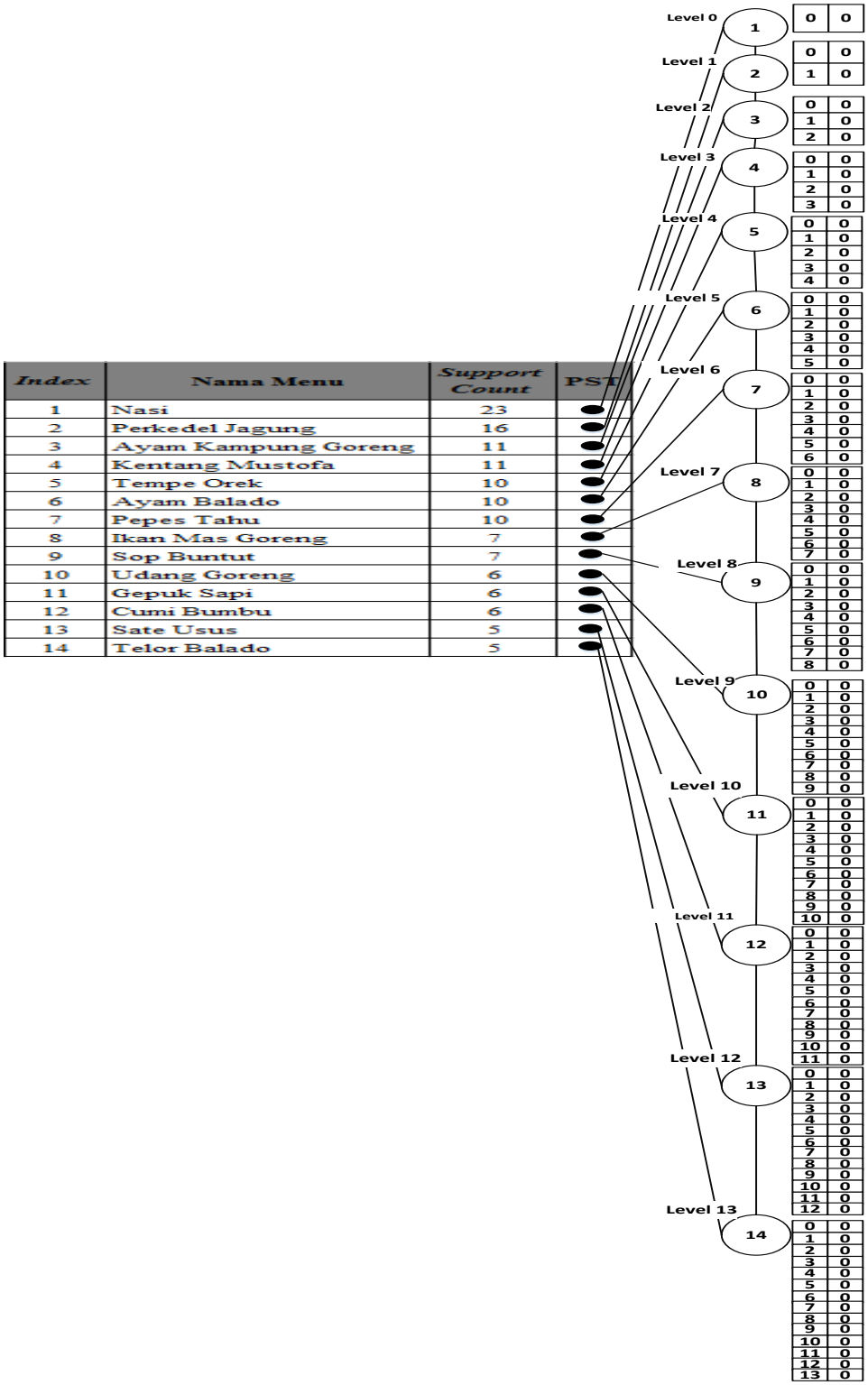
NomorBill	*Id Global Item*
151200026	12
151200027	1
151200027	4
151200027	8
151200027	13

e. Pembentukan *CFP-Tree*

Setelah proses *mapping* selesai kemudian ke tahap selanjutnya adalah *insert to CFP-Tree* yaitu memasukan setiap data transaksi ke dalam *tree*, berikut merupakan tahapanya:

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.*

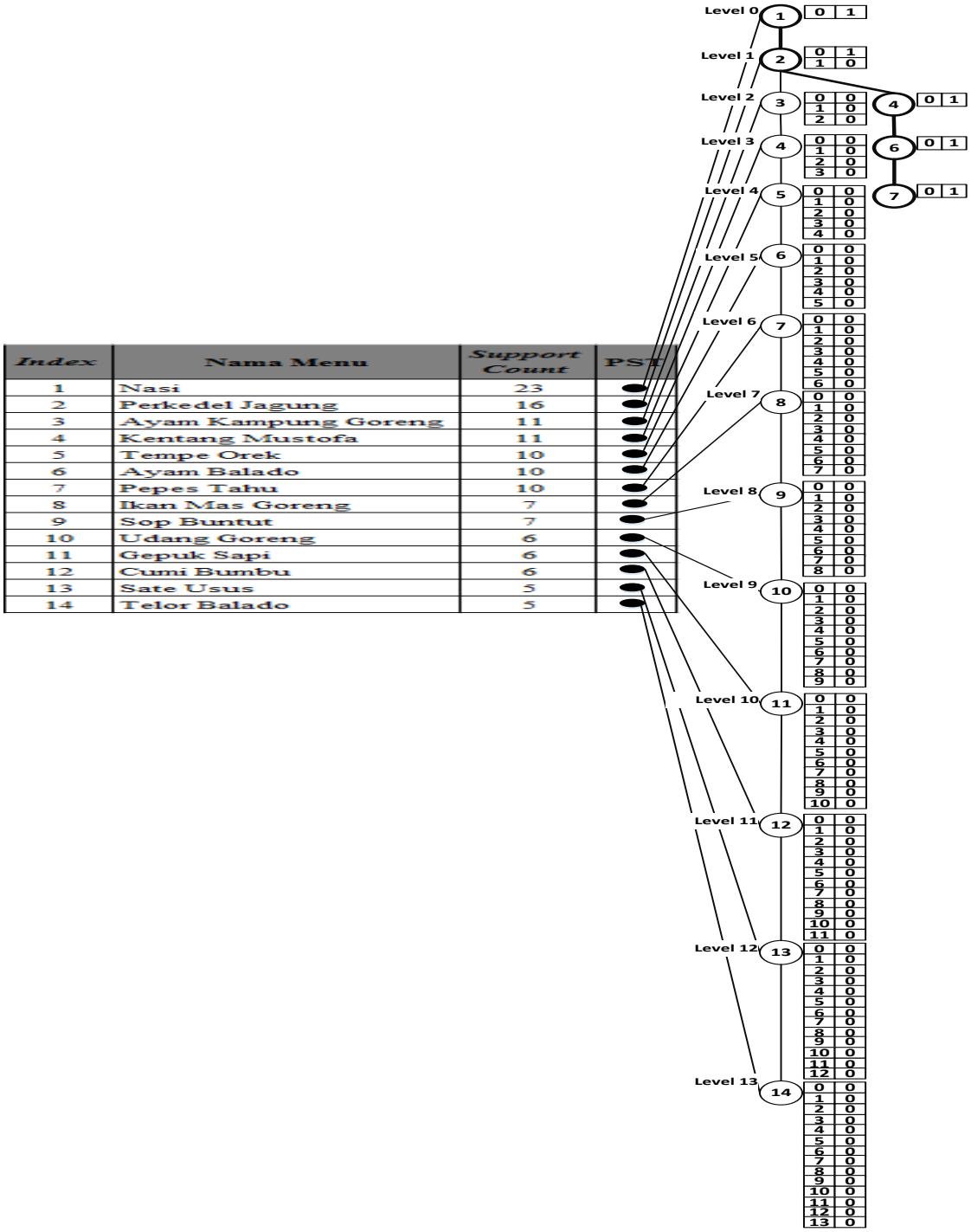
Berikut merupakan hasil pembentukan Left Branch dapat dilihat pada gambar 3.2 :



Index	Nama Menu	*Support Count*	PST
1	Nasi	23	
2	Perkedel Jagung	16	
3	Ayam Kampung Goreng	11	
4	Kentang Mustofa	11	
5	Tempe Orek	10	
6	Ayam Balado	10	
7	Pepes Tahu	10	
8	Ikan Mas Goreng	7	
9	Sop Buntut	7	
10	Udang Goreng	6	
11	Gepuk Sapi	6	
12	Cumi Bumbu	6	
13	Sate Usus	5	
14	Telor Balado	5	

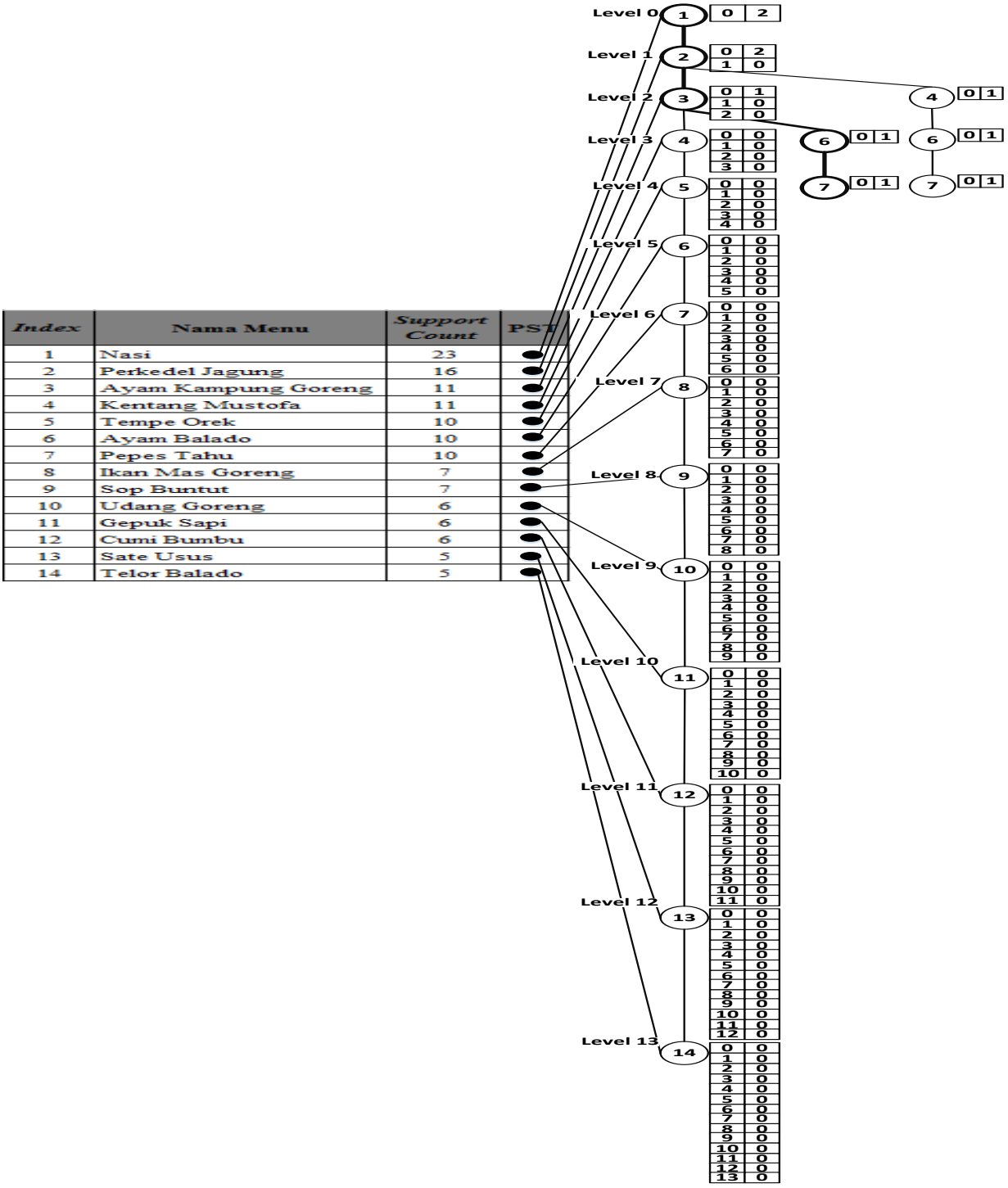
Gambar 3. 2 *Left Branch*

- Untuk pembentukan *CFP-Tree* berdasarkan data transaksi yang pertama yaitu dengan Nomor Bill 151200001 dengan *id global item*: 1, 2, 4, 6, dan 7 merupakan data awal yang akan diproses. Pembentukan *CFP-Tree* dengan Nomor Bill 151200001 dapat dilihat pada gambar 3.3 sebagai berikut:



Gambar 3. 3 *CFP-Tree* Nomor Bill 151200001

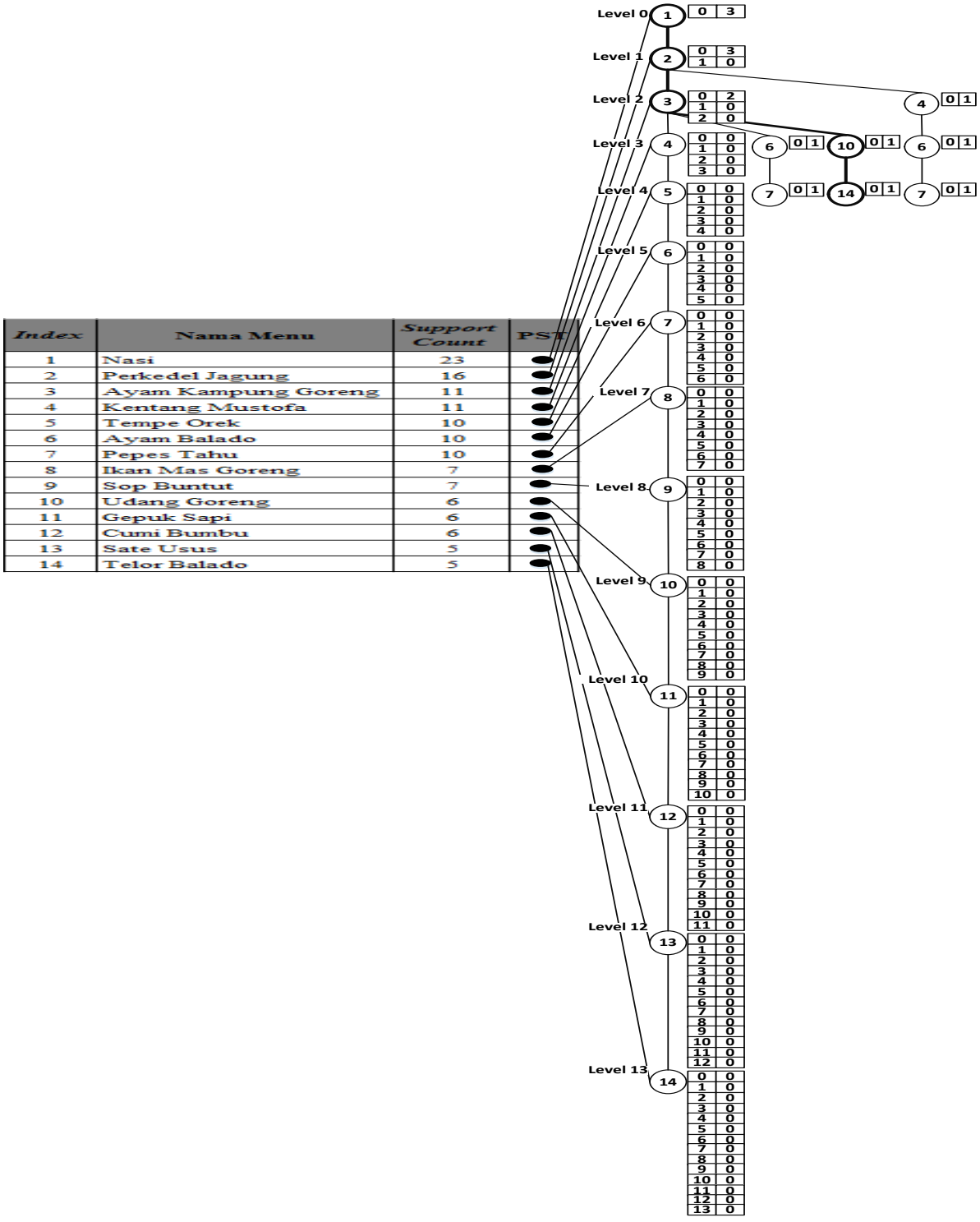
- Pembentukan *CFP-Tree* berdasarkan data transaksi kedua yaitu dengan Nomor Bill 151200002 dengan *id global item*: 1, 2, 3, 6, dan 7 akan membentuk *CFP-Tree* yang dapat dilihat pada gambar 3.3 sebagai berikut:



Index	Nama Menu	*Support Count*	PST
1	Nasi	23	
2	Perkedel Jagung	16	
3	Ayam Kampung Goreng	11	
4	Kentang Mustofa	11	
5	Tempe Orek	10	
6	Ayam Balado	10	
7	Pepes Tahu	10	
8	Ikan Mas Goreng	7	
9	Sop Buntut	7	
10	Udang Goreng	6	
11	Gepuk Sapi	6	
12	Cumi Bumbu	6	
13	Sate Usus	5	
14	Telor Balado	5	

Gambar 3. 4 *CFP-Tree* Nomor Bill 151200002

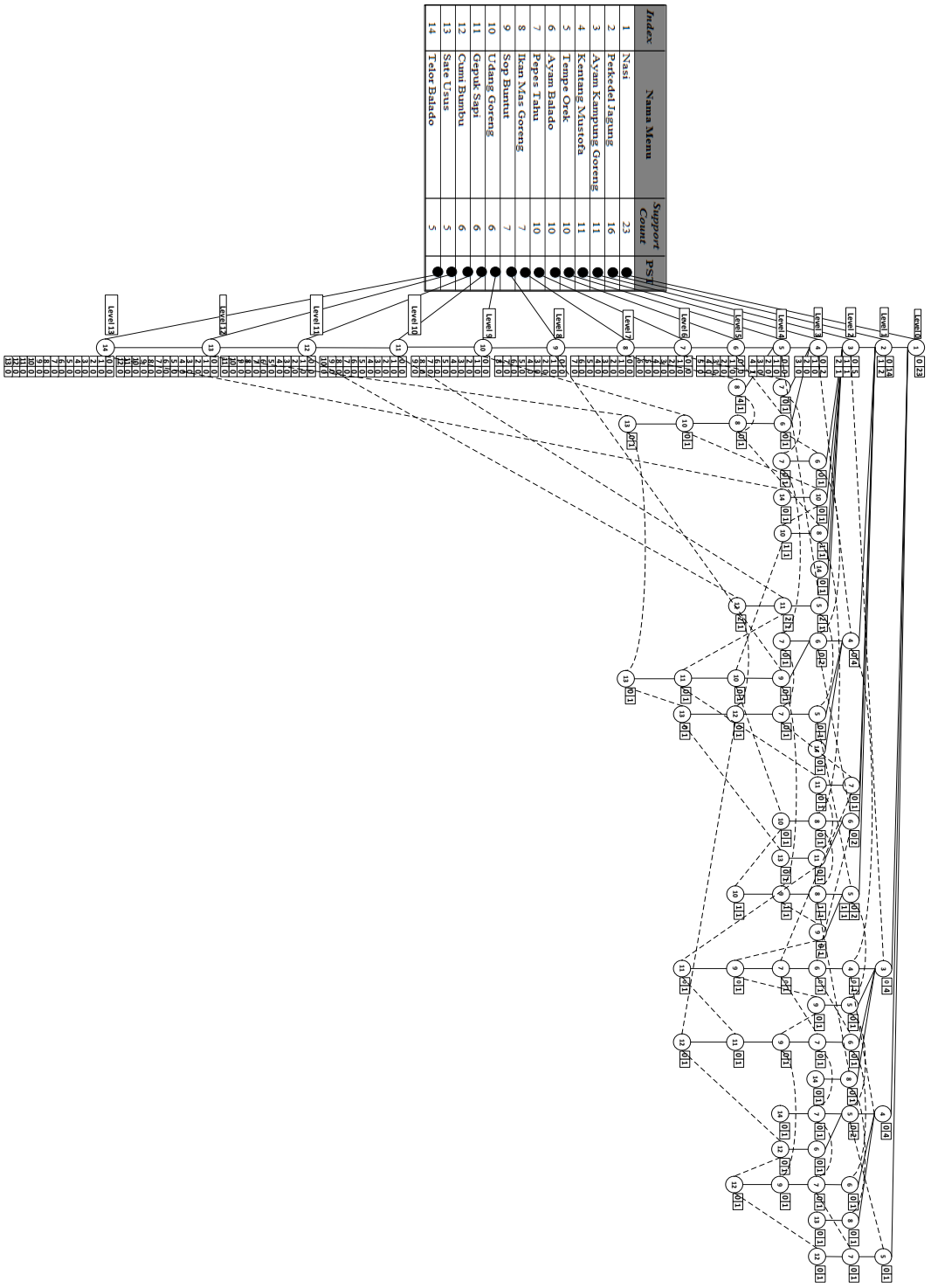
- Pembentukan *CFP-Tree* berdasarkan data transaksi ketiga yaitu dengan Nomor Bill 151200003 dengan *id global item*: 1, 2, 3, 10, dan 14 akan membentuk *CFP-Tree* yang dapat dilihat pada gambar 3.4 sebagai berikut:



Index	Nama Menu	*Support Count*	PST
1	Nasi	23	
2	Perkedel Jagung	16	
3	Ayam Kampung Goreng	11	
4	Kentang Mustofa	11	
5	Tempe Orek	10	
6	Ayam Balado	10	
7	Pepes Tahu	10	
8	Ikan Mas Goreng	7	
9	Sop Buntut	7	
10	Udang Goreng	6	
11	Gepuk Sapi	6	
12	Cumi Bumbu	6	
13	Sate Usus	5	
14	Telor Balado	5	

Gambar 3. 5 *CFP-Tree* Nomor Bill 15120000

-Untuk proses pembentukan *CFP-Tree* selanjutnya dapat dilihat pada gambar D-4 lampiran D. Berikut adalah *Global CFP-Tree* yang terbentuk dari seluruh transaksi, yang diacu pada gambar 3.6 sebagai berikut:



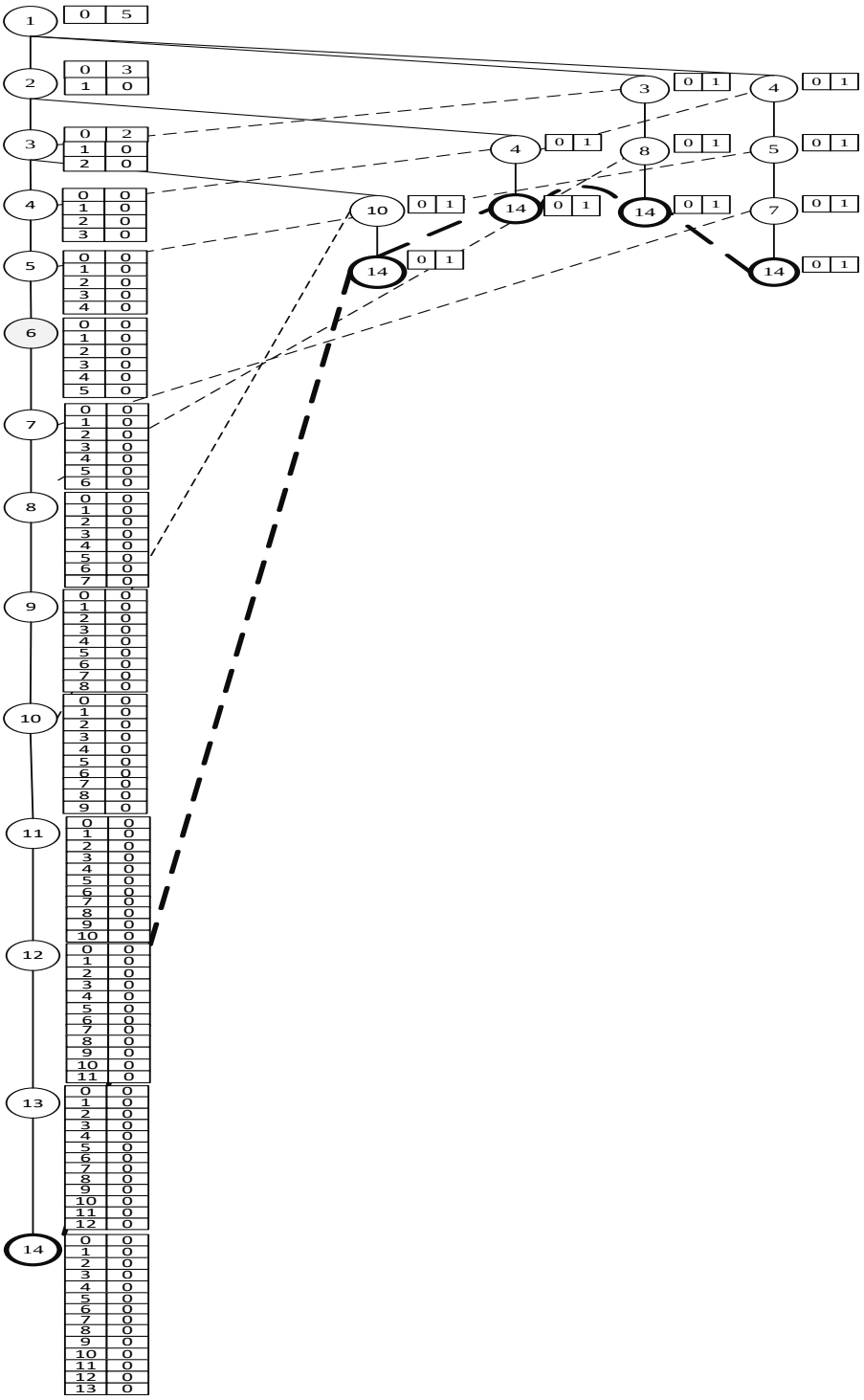
Gambar 3. 6 *Global CFP-Tree*

f. Penggalian *Frequent Patterns*

Setelah tahap pembangunan *CFP-Tree* dari sekumpulan data transaksi, akan dilakukan penggalian *Frequent Patterns* dengan ketentuan sebagai berikut:

1. Dimulai dengan memilih item dengan *support count* terkecil atau *Index* terakhir dari *Global* item *table*.
2. Lakukan pencarian *node* yang berkaitan dengan item terkecil yang telah dipilih dan menghitung *count* = 1 pada setiap *node* di mulai dari path sampai ke *root*.
3. Masukan setiap item yang memenuhi *minimum support* ke *local item table.*
4. Berikan *Index* baru pada *node* yang ada di *local item table* secara *ascending*.
5. Setelah *local item table* terbentuk lakukan pemetaan *index* baru terhadap *node* yang memenuhi *support* sesuai dengan index di *local item table*
6. Setelah dilakukan pemetaan lakukan pembentukan *Local CFP-Tree* dengan aturan yang sama saat pembentukan Global *CFP-Tree*
7. Lakukan pembentukan *Local CFP-Tree projection* yaitu dimana item yang di pilih akan menjadi *root* dan item yang ada di *local CFP-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 ke-i

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 14. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 14, untuk jelas nya dapat dilihat pada gambar berikut:



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

Berikut hasil penelusuran untuk *index* 14 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.11.

Tabel 3. 11 Penelusuran *Index* 14

No	penelusuran
1	14,10,3,2,1
2	14,3,2,1
3	14,4,2,1
4	14,8,3,1
5	14,7,5,4,1

Dari gambar 3.7 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(5), 2(3), 3(3), 4(2), 5(1), 7(1), 8(1), 10(1) karena *minimum supportnya* adalah 5, maka *node* yang memenuhi minimum *support* adalah 1(5). Untuk *node* yang tidak memenuhi minimum *suppor*t, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.12.

Tabel 3. 12 *Local Item Table index* 14

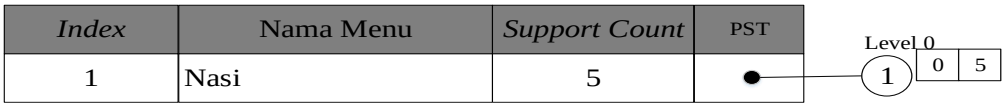
Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	5

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 3.15

Tabel 3.13 Hasil *Mapping Local CFP-Tree*

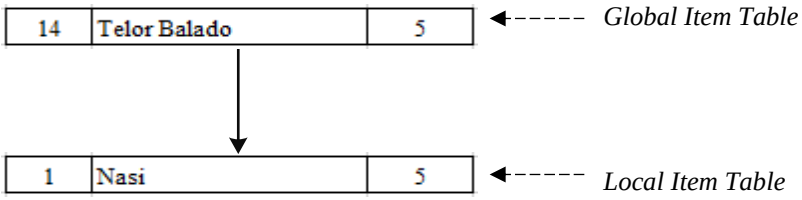
No	penelusuran
1	1
2	1
3	1
4	1
5	1

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:



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

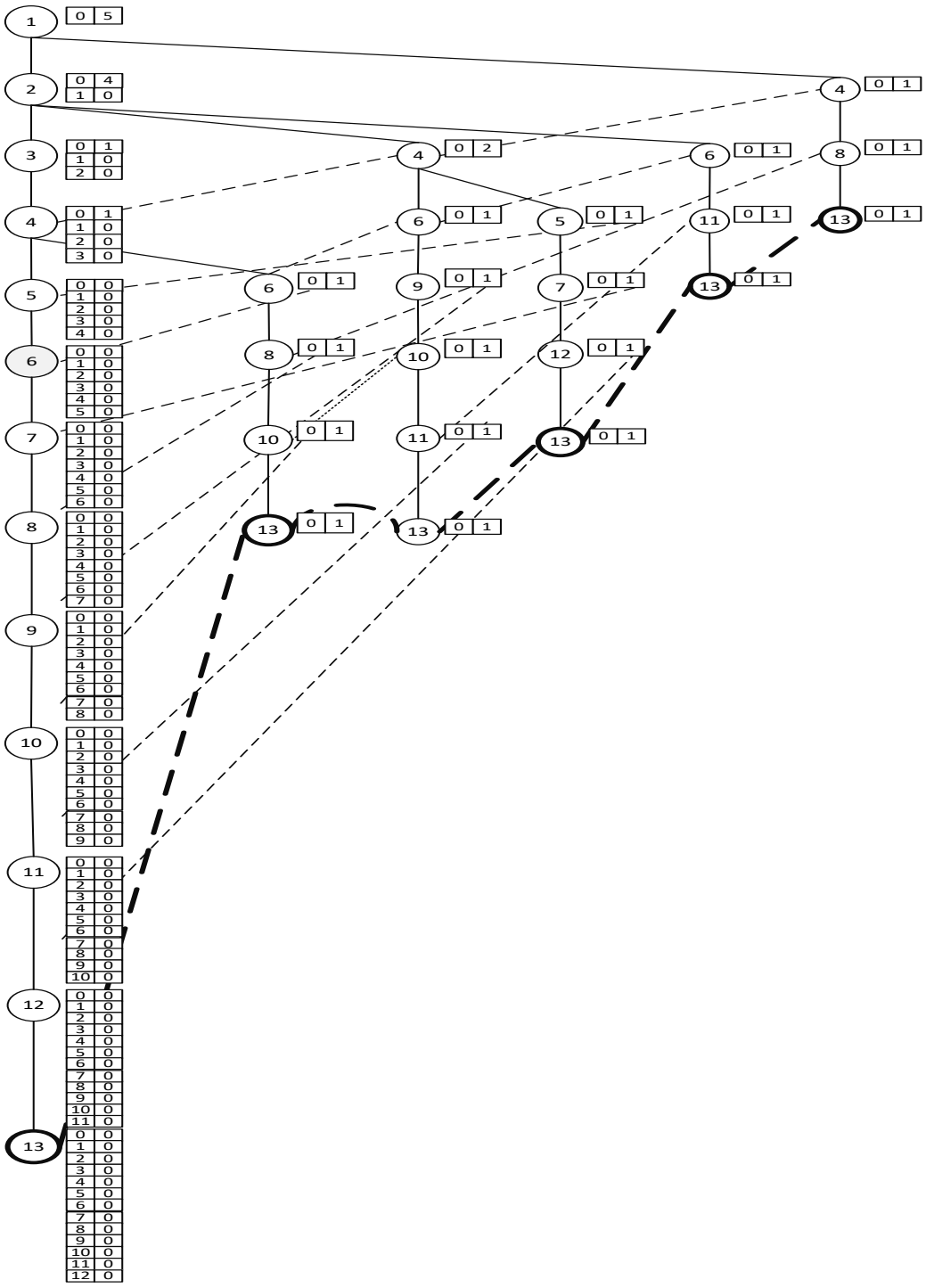
Setelah *Local CFP-Tree index* 14 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 14, dimana *index* 14 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 14 Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 14 mulai dari *path* sampai *root* secara *battom-up*, untuk lebih jelasnya dapat dlihat pada gambar berikut:



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 14 dengan Nama Menu Telor Balado adalah (14,1:5)

- Untuk pencariian *Local Frequent Pattern Tree* untuk *index* 13. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 13, untuk jelas nya dapat dilihat pada gambar berikut:



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

Berikut hasil penelusuran untuk *index* 13 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.14.

Tabel 3.14 Penelusuran *Index* 13

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

Dari gambar 3.10 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(5), 2(4), 3(1), 4(4), 5(1), 6(3), 7(1), 8(2), 9(1), 10(2), 11(2),12(1) karena *minimum supportnya* adalah 5, maka *node* yang memenuhi minimum *support* adalah 1(5). untuk *node* yang tidak memenuhi minimum *support*, maka tidak dimasukan ke dalam *local item* table. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.15

Tabel 3. 15 *Local Item Table index* 13

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	5

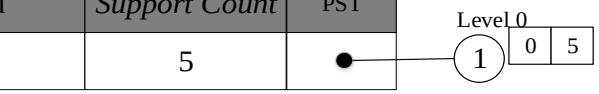
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pad tabel 3.16

Tabel 3. 16 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1
3	1
4	1
5	1

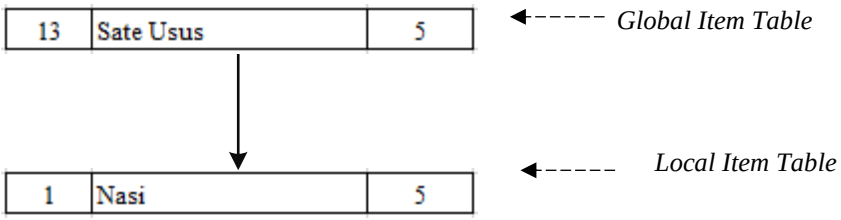
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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 13:

Index	Nama Menu	*Support Count*	PST
1	Nasi	5	



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

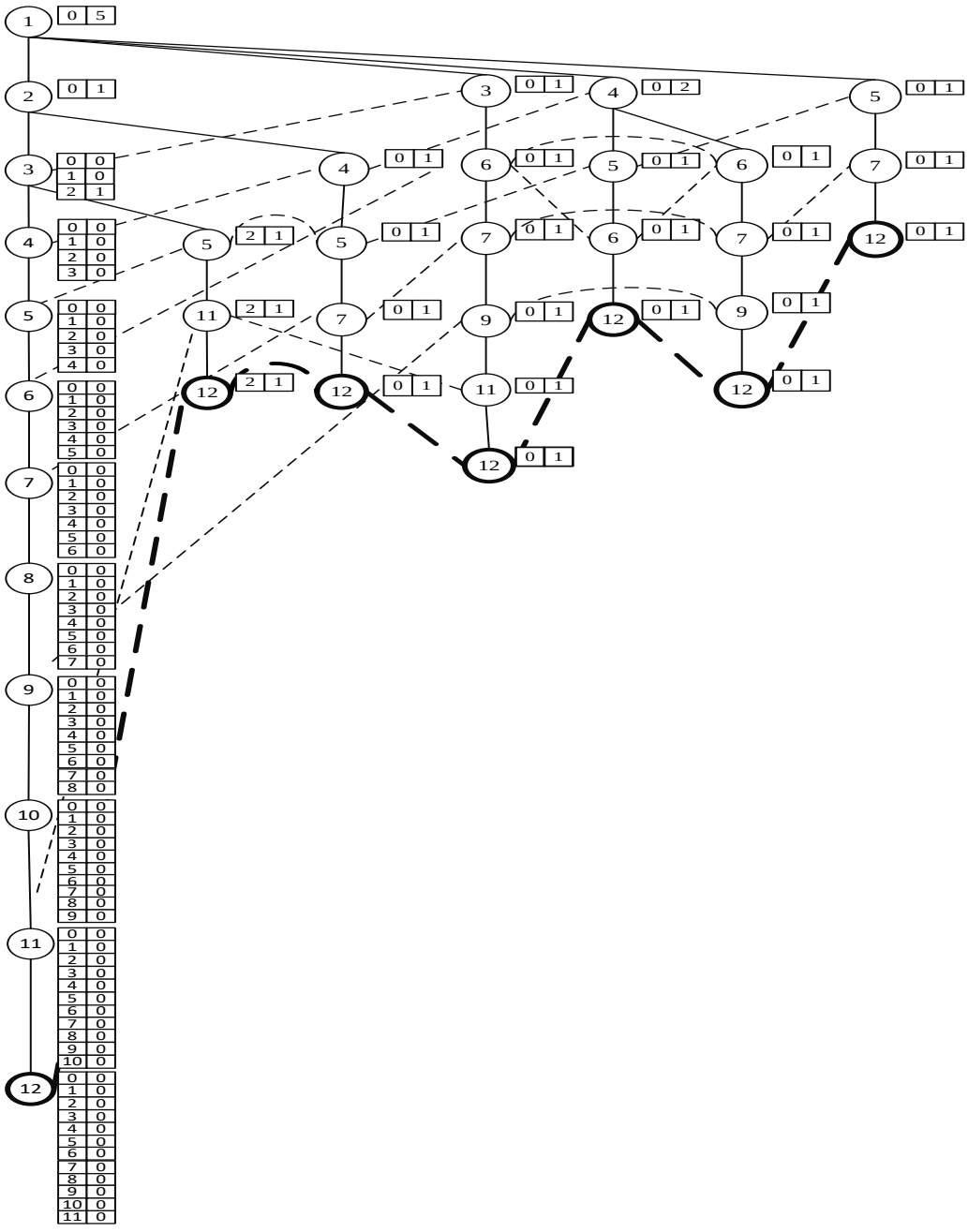
Setelah *Local CFP-Tree index* 13 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 13, dimana *index* 13 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 13. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 13 mulai dari *path* sampai *root* secara *battom-up*, untuk lebih jelasnya dapat dlihat pada gambar berikut:



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 13 dengan Nama Menu Sate Usus adalah sebagai berikut: (13,1:5).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 12. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 12, untuk jelas nya dapat dilihat pada gambar 3.13 :



Gambar 3.13 *CFP-Tree* Berakhiran *index* 12

Berikut hasil penelusuran untuk *index* 12 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.17.

Tabel 3. 17 Penelusuran *Index* 12

No	penelusuran
1	12,11,5,3
2	12,7,5,4,2,1
3	12,11,9,7,6,3,1
4	12,6,5,4,1
5	12,9,7,6,4,1
6	12,7,5,1

Dari gambar 3.13 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(5), 2(1), 3(2), 4(3), 5(4), 6(3), 7(4), 9(2), 11(2) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi minimum *support* adalah 1(5) . untuk *node* yang tidak memenuhi minimum support, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut merupakan *local item table* yang terbentuk pada tabel 3.18:

Tabel 3. 18 *Local Item Table index* 12

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	5

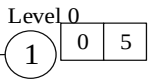
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.19

Tabel 3.19 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1
3	1
4	1
5	1

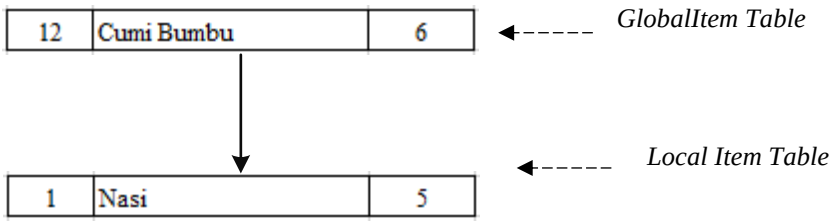
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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 12:

Index	Nama Menu	*Support Count*	PST
1	Nasi	5	



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

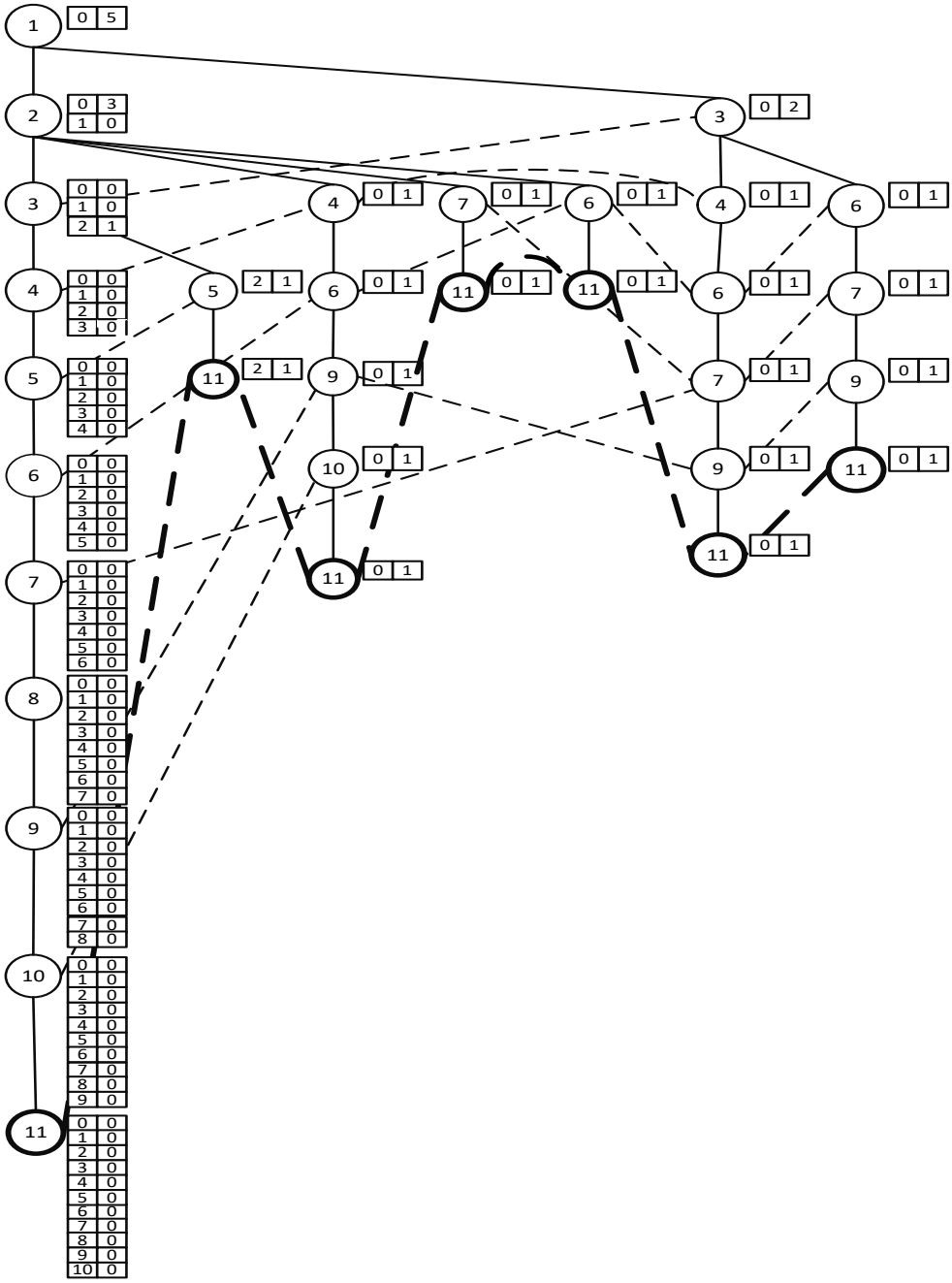
Setelah *Local CFP-Tree index* 10 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 10, dimana *index* 10 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 10. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di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



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 12 dengan Nama Menu Cumi Bumbu adalah sebagai berikut: (12,1:5).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 11. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 11, untuk jelas nya dapat dilihat pada gambar gambar 3.16 :



Gambar 3.16 *CFP-Tree* Berakhiran *index* 11

Berikut hasil penelusuran untuk *index* 11 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.20

Tabel 3.20 Penelusuran *Index* 11

No	Penelusuran
1	11,5,3
2	11,10,9,6,4,2,1
3	11.7.2.1
4	11,6,2,1
5	11,9,7,6,4,3,1
6	11,9,7,6,3,1

Dari gambar 3.16 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(5), 2(3), 3(3), 4(2), 5(1), 6(4), 7(3), 9(3), 10(1) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(5). Untuk *node* yang tidak memenuhi minimum *support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.21

Tabel 3. 21 *Local Item Table index* 11

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	5

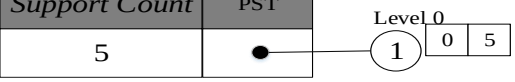
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.22

Tabel 3.22 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1
3	1
4	1
5	1

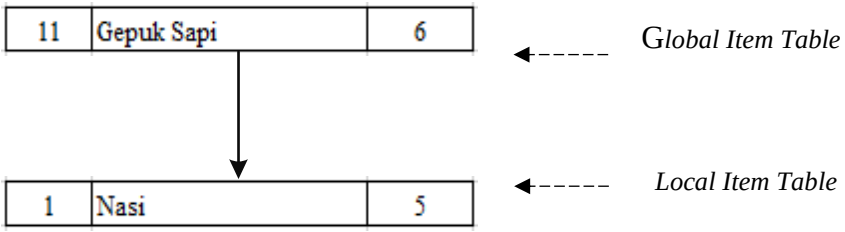
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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 11:

Index	Nama Menu	*Support Count*	PST
1	Nasi	5	



Gambar 3.17 *Local CFP-Tree index* 11

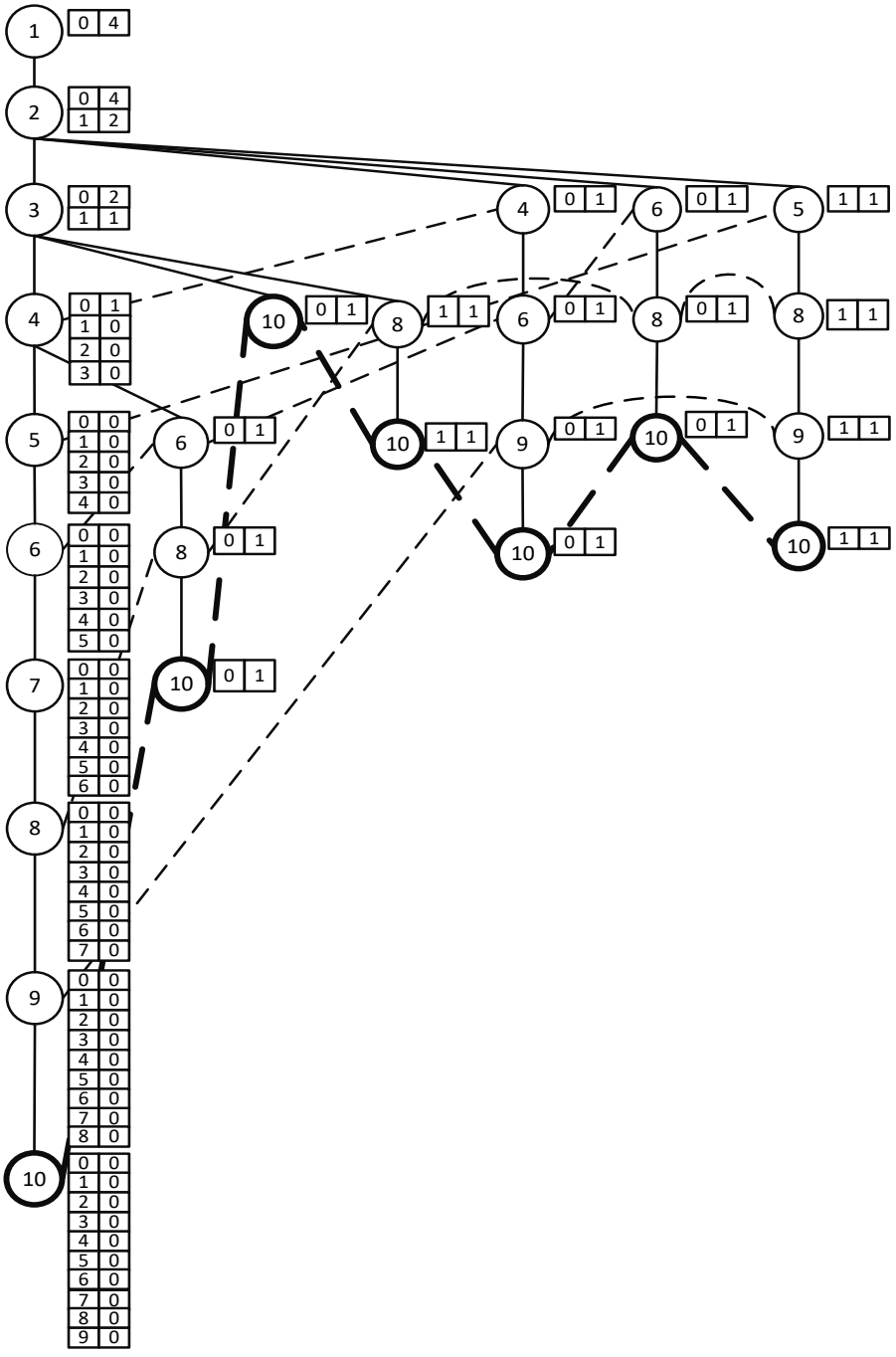
Setelah *Local CFP-Tree index* 11 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 11, dimana *index* 11 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 11. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di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



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 11 dengan Nama Menu Gepuk Sapi adalah sebagai berikut: (11,1:5).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 10. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 10, untuk jelas nya dapat dilihat pada gambar gambar 3.19 :



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

Berikut hasil penelusuran untuk *index* 10 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.23.

Tabel 3.23 Penelusuran *Index* 10

No	Penelusuran
1	10,8,6,4,3,2,1
2	10,3,2,1
3	10,8,3,2
4	10,9,6,4,2,1
5	10,8,6,2,1
6	10,,8,5,2

Dari gambar 3.19 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(4), 2(6), 3(3), 4(2), 5(1), 6(3), 8(4), 9(2) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 2(6). Untuk yang tidak memenuhi minimum *support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.24

Tabel 3. 24 *Local Item Table index* 10

Index	*Node*	Nama Menu	*Support Count*
1	2	Perkedel Jagung	6

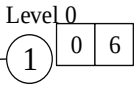
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.25.

Tabel 3. 25 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1
3	1
4	1
5	1
6	1

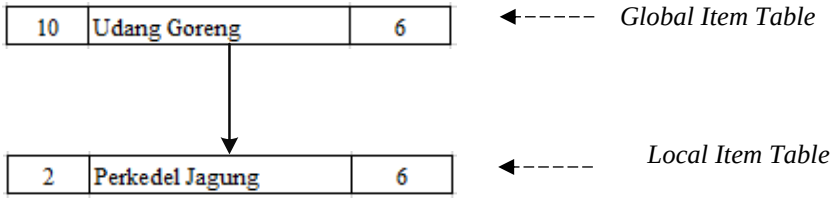
Dari *Local Item Table* yang didapatkan pada tabel diatas dibuat *Local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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 10:

Index	Nama Menu	*Support Count*	PST
2	Perkedel Jagung	6	



Gambar 3. 20 *Local CFP-Tree index* 10

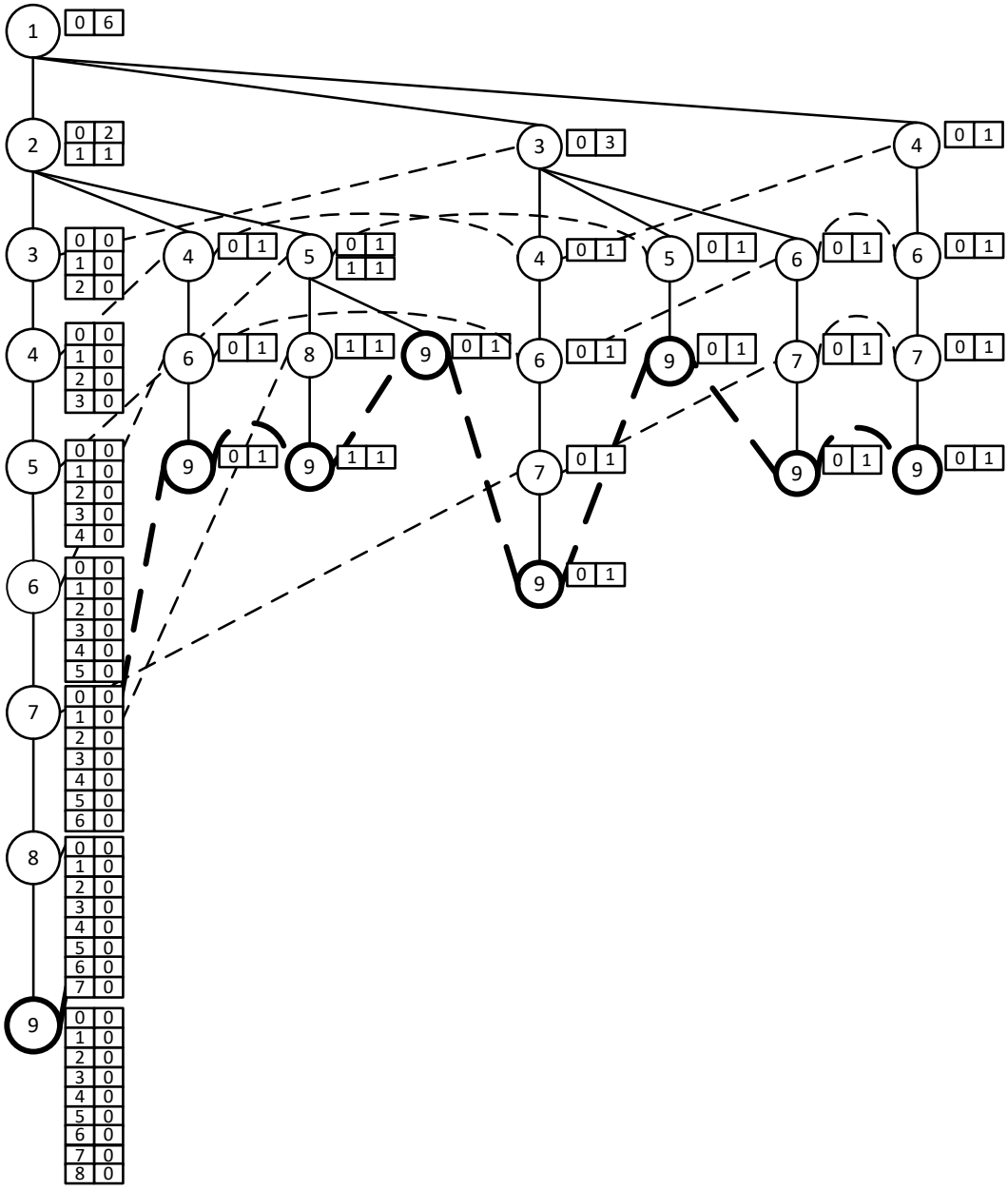
Setelah *Local CFP-Tree index* 10 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 10, dimana *index* 10 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 10. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di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



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 10 dengan Nama Menu Udang Goreng adalah sebagai berikut: (10,2:6).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 9. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 9, untuk jelas nya dapat dilihat pada gambar gambar 3.22:



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

Berikut hasil penelusuran untuk *index* 9 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.26.

Tabel 3.26 Penelusuran *Index* 9

No	Penelusuran
1	9,6,4,2,1
2	9,8,5,2
3	9,5,2,1
4	9,7,6,4,3,1
5	9,5,3,1
6	9,7,6,3,1
7	9,7,6,4,1

Dari gambar 3.22 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(6), 2(3), 3(3), 4(3), 6(4), 7(3),8(1) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(6). Untuk *node* yang tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.27.

Tabel 3.27 *Local Item Table index* 9

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	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 3.28.

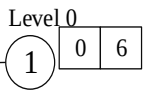
Tabel 3.28 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1

No	Penelusuran
2	1
3	1
4	1
5	1
6	1

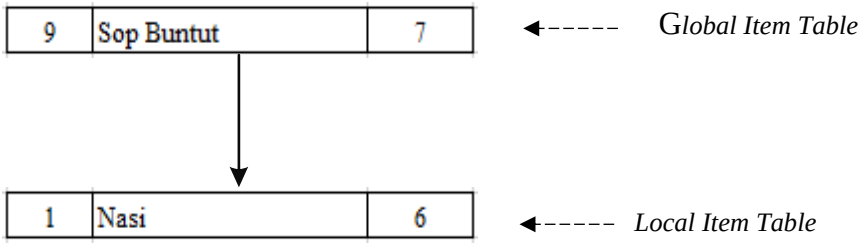
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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 9:

Index	Nama Menu	*Support Count*	PST
1	Nasi	6	



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

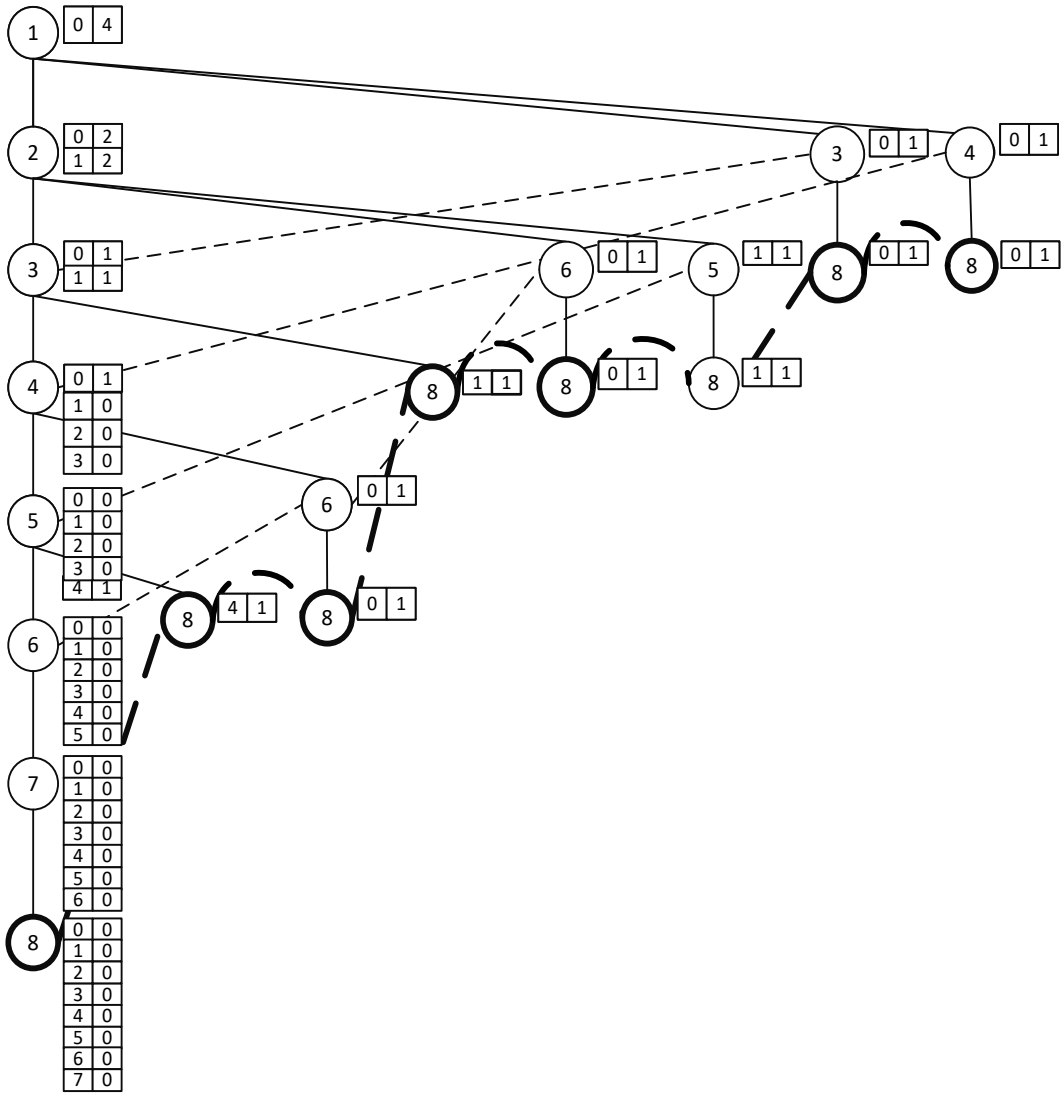
Setelah *Local CFP-Tree index* 9 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 7, dimana *index* 9 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 9. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 9 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.24:



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

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 9 dengan Nama Menu Sop Buntut adalah sebagai berikut: (9,1:6).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 8. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran 8, untuk jelas nya dapat dilihat pada gambar gambar 3.25 :



Gambar 3.25 *CFP-Tree* Berakhiran *index* 8

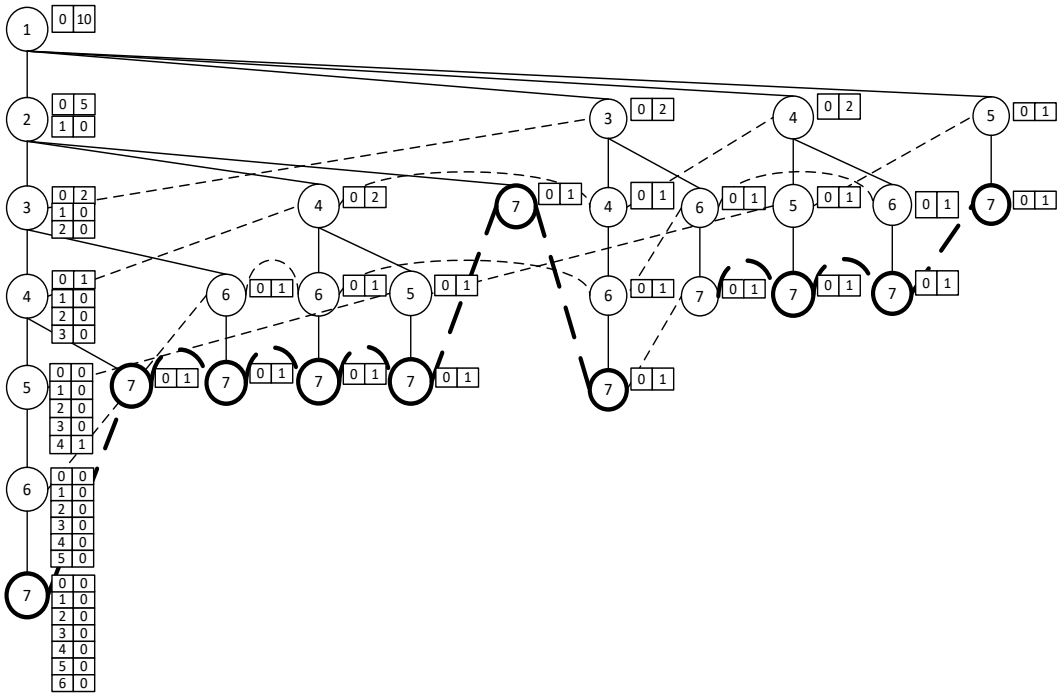
Berikut hasil penelusuran untuk *index* 8 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.29.

Tabel 3.29 Penelusuran *Index* 8

No	Penelusuran
1	8,5
2	8,6,3,2,1
3	8,3,2
4	8,6,2,1
5	8,5,2
6	8,3,1
7	8,4,1

Dari gambar 3.25 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(4), 2(4), 3(3), 4(2),5(2), 6(2) karena *minimum support*-nya adalah 5, maka tidak dimasukan ke dalam *local item table*. Pada *node* diatas tidak ada yang memenuhi nilai *minimum support*, maka tidak terbentuk *local item table* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

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



Gambar 3. 26 *CFP-Tree* Berakhiran *index* 7

Berikut hasil penelusuran untuk *index* 7 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.30

Tabel 3.30 Penelusuran *Index* 7

No	Penelusuran
1	7,4,3,2,1
2	7,6,3,2,1
3	7,6,4,2,1
4	7,5,4,2,1
5	7,2,1
6	7,6,4,3,1
7	7,6,3,1
9	7,5,4,1
No	Penelusuran
10	7,6,4,1
11	7,5,1

Dari gambar 3.26 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(10), 2(5), 3(4), 4(6), 5(3), 6(5) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(10), 2(5), 4(6), 6(5). Untuk *node* 2(5), yang tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.31.

Tabel 3. 31 *Local Item Table index* 7

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	10
2	4	Kentang Mustofa	6
3	2	Perkedel Jagung	5
4	6	Ayam Balado	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 3.32.

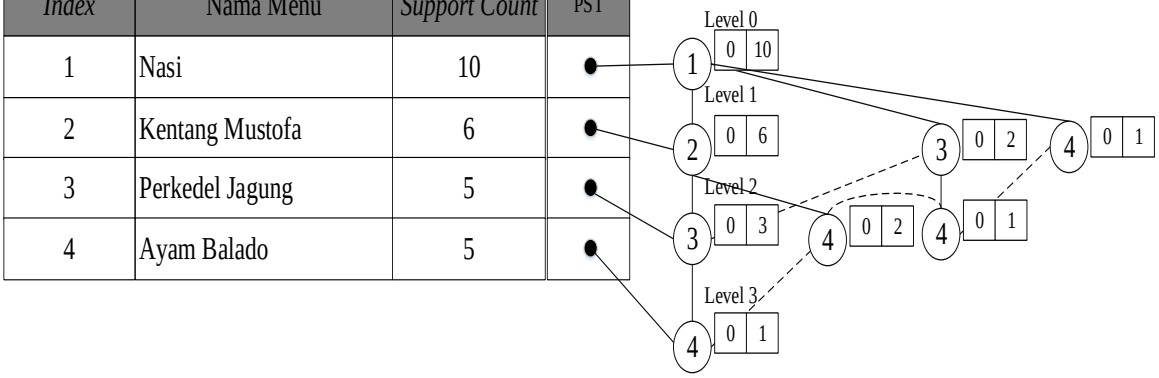
Tabel 3.32 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1,2,3
2	1,3,4
3	1,2,3,4
4	1,2,3
5	1,3
No	Penelusuran
6	1,2,4
7	1,4

8	1,2
9	1,2,4
10	1

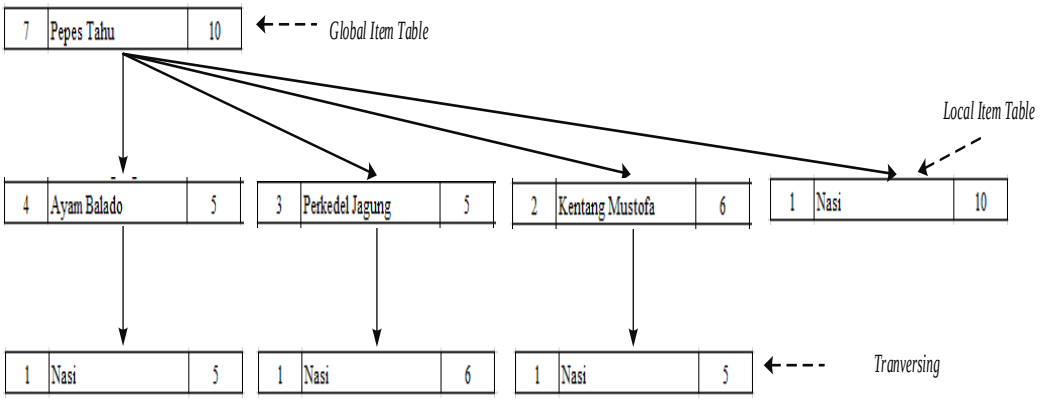
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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* 7:

Index	Nama Menu	*Support Count*	PST
1	Nasi	10	
2	Kentang Mustofa	6	
3	Perkedel Jagung	5	
4	Ayam Balado	5	



Gambar 3. 27 *Local CFP-Tree index* 7

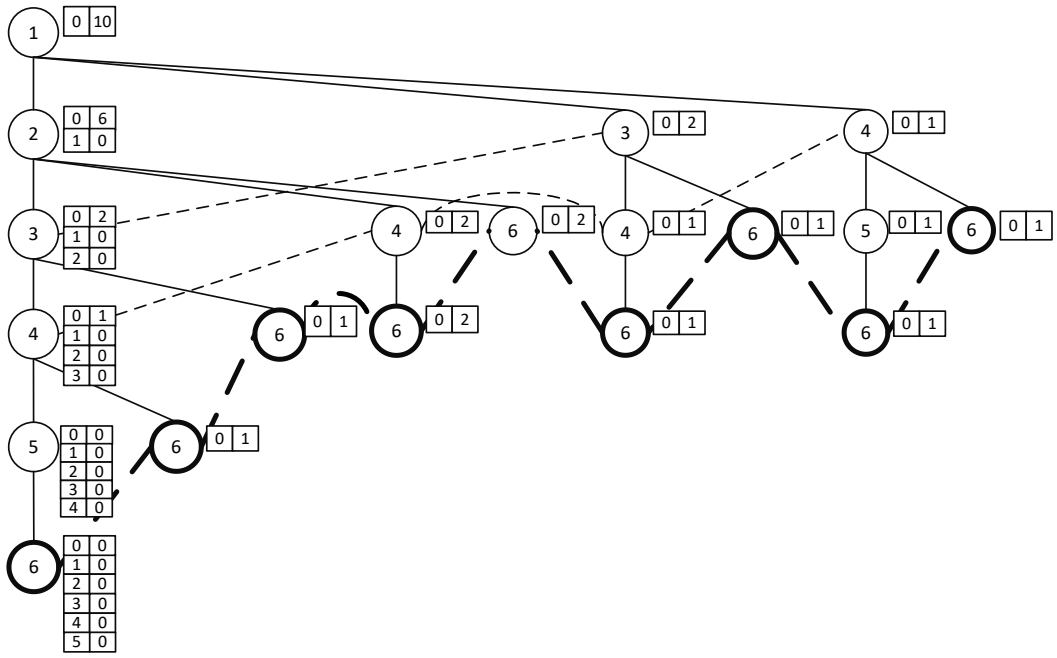
Setelah *Local CFP-Tree index* 7 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 7, dimana *index* 7 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 7. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 7 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.28:



Gambar 3. 28 *Local CFP-Tree Projection* 7

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 7 dengan Nama Menu Pepes Tahu adalah sebagai berikut: (7,4:5), (7,4,1:5), (7,3:5), (7,3,1:6), (7,2:6), (7,2,1:5), (7,1:10)

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 6. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran 6, untuk jelas nya dapat dilihat pada gambar berikut:



Gambar 3. 29 *CFP-Tree* Berakhiran *index* 6

Berikut hasil penelusuran untuk *index* 6 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.33

Tabel 3.33 Penelusuran *Index* 6

No	penelusuran
1	6,4,3,2,1
2	6,3,2,1
3	6,4,2,1
4	6,4,2,1
5	6,2,1
6	6,4,3,1
7	6,3,1
8	6,5,4,1
9	6,4,1

Dari gambar 3.29 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(10), 2(6), 3(4), 4(5), 5(1) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(10), 2(6), 4(5). Untuk *node* yang tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.34.

Tabel 3.34 *Local Item Table index* 6

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	10
2	2	Perkedel Jagung	6
3	4	Kentang Mustofa	5

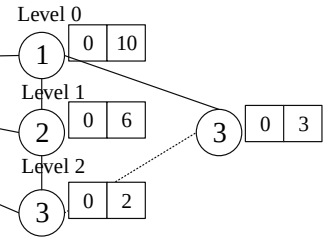
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.35.

Tabel 3.35 Hasil *Mapping Local CFP-Tree*

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

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* 6:

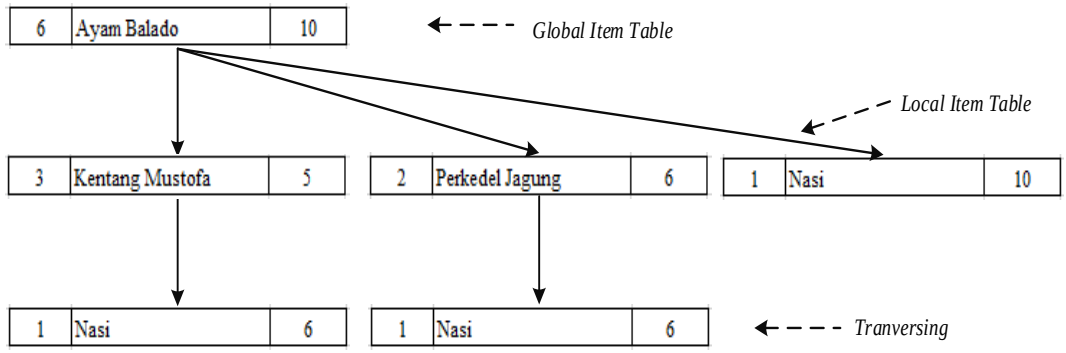
Index	Nama Menu	*Support Count*	PST
1	Nasi	10	●
2	Perkedel Jagung	6	●
3	Kentang Mustofa	5	●



Gambar 3.30 *Local CFP-Tree index* 6

Setelah *Local CFP-Tree index* 6 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 6, dimana *index* 6 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 6. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di

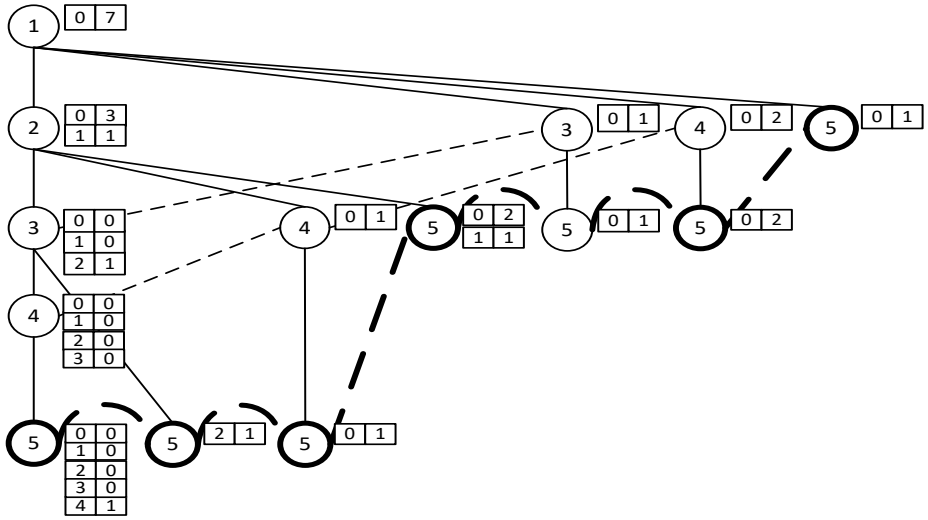
index 6 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.31:



Gambar 3. 31 *Local CFP-Tree Projection* 6

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 6 dengan Nama Menu Ayam Balado adalah sebagai berikut: (6,3:5), (6,3,1:6), (6,2:6), (6,2,1:6) dan (6,1:10).

- Untuk Pencarian *Local Frequent Pattern Tree* untuk *index* 5. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran 5, untuk jelas nya dapat dilihat pada gambar 3.32:



Gambar 3. 32 *CFP-Tree* Berakhiran *index* 5

Berikut hasil penelusuran untuk *index* 5 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.36.

Tabel 3.36 Penelusuran *Index* 5

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

Dari gambar 3.32 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(7), 2(4), 3(2), 4(2) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(7). Untuk yang *node* tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.37

Tabel 3.37 *Local Item Table index* 5

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	7

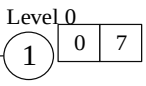
Setelah *local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai minimum *support* dengan nomor *index* yang sudah diberikan. Berikut hasil *mapping* yang dapat dilihat pada tabel 3.38.

Tabel 3.38 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1
3	1
4	1
5	1
6	1
7	1

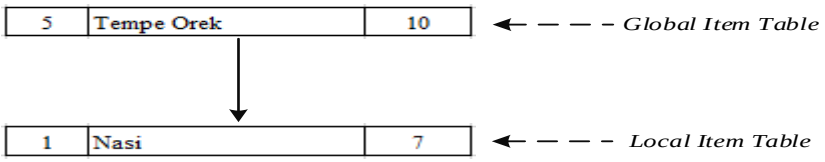
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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* 5:

Index	Nama Menu	*Support Count*	PST
1	Nasi	7	



Gambar 3.33 *Local CFP-Tree index* 5

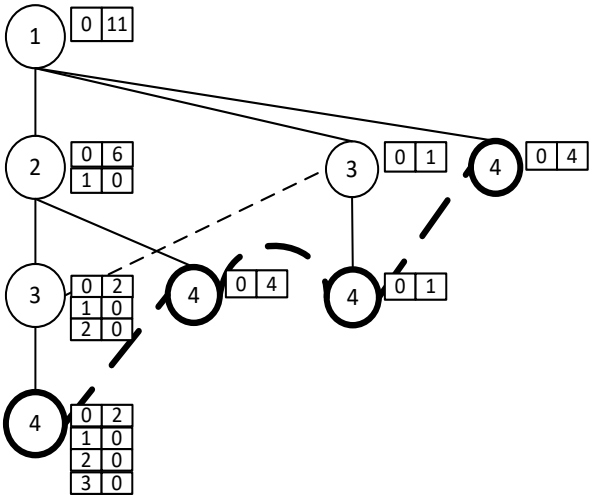
Setelah *Local CFP-Tree index* 5 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 5, dimana *index* 5 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 5. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 5 mulai dari *path* sampai *root* secara *battom-up*, untuk lebih jelasnya dapat dlihat pada gambar 3.34:



Gambar 3. 34 *Local CFP-Tree Projection* 5

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 5 dengan Nama Menu Tempe Orek adalah sebagai berikut: (5,1:7).

- Pencarian *Local Frequent Pattern Tree* untuk *index* 4. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran 4, untuk jelas nya dapat dilihat pada gambar berikut:



Gambar 3.35 *CFP-Tree* Berakhiran *index* 4

Berikut hasil penelusuran untuk *index* 4 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.39.

Tabel 3.39 Penelusuran *Index* 4

No	Penelusuran
1	4,3,2,1
2	4,3,2,1
3	4,2,1
4	4,2,1
5	4,2,1
6	4,2,1
7	4,3,1
8	4,1

No	Penelusuran
9	4,1
10	4,1
11	4,1

Dari gambar 3.35 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(11), 2(6), 3(3) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(11), 2(6). Untuk *node* yang tidak memenuhi *minimum support*, maka tidak dimasukan ke dalam *local item table*. Setelah itu diberikan *index* yaitu penomoran secara *ascending* dari jumlah *support count* terbesar ke terkecil. Berikut *local item table* yang terbentuk pada tabel 3.40

Tabel 3.40 *Local Item Table index* 4

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	11
2	2	Perkedel Jagung	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 3.41.

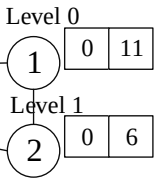
Tabel 3.41 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1,2
2	1,2
3	1,2
4	1,2
5	1,2
6	1,2
7	1
8	1

No	Penelusuran
9	1
10	1
11	1

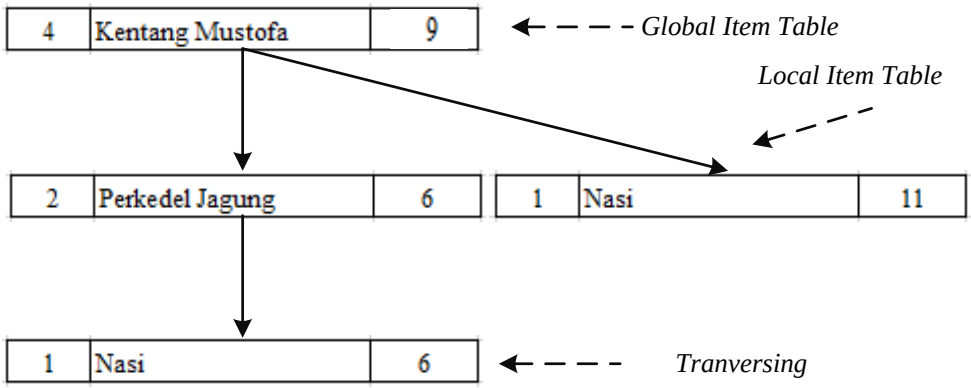
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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* 4:

Index	Nama Menu	*Support Count*	PST
1	Nasi	11	
2	Perkedel Jagung	6	



Gambar 3. 36 *Local CFP-Tree index* 4

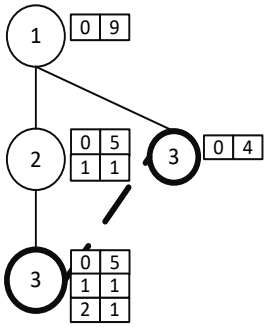
Setelah *Local CFP-Tree index* 4 terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 4, dimana *index* 4 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 4. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 4 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.37:



Gambar 3. 37 *Local CFP-Tree Projection* 4

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 4 dengan Nama Menu Kentang Mustofa adalah sebagai berikut: (4,2:6), (4,2,1:6), dan (4,1:11).

- Pencarian *Local Frequent Pattern Tree* untuk *index* 3. Cari jalur (*node*) pada *Global CFP-Tree* yang hanya berakhiran 3, untuk jelas nya dapat dilihat pada gambar 3.38:



Gambar 3.38 *CFP-Tree* Berakhiran *index* 3

Berikut hasil penelusuran untuk *index* 3 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.42.

Tabel 3.42 Penelusuran *Index* 3

No	Penelusuran
1	3
2	3,2
3	3,2,1

No	Penelusuran
4	3,2,1
5	3,2,1
6	3,2,1
7	3,2,1
8	3,1
9	3,1
10	3,1
11	3,1

Dari gambar 3.38 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(9), 2(6) karena *minimum support*-nya adalah 5, maka *node* yang memenuhi *minimum support* adalah 1(9), 2(6). Maka semua *node* dimasukan ke dalam *local item table*. Berikut *local item table* yang terbentuk pada tabel 3.43.

Tabel 3.43 *Local Item Table index* 3

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	9
2	2	Perkedel Jagung	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 3.44.

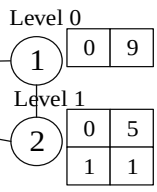
Tabel 3.44 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	2
2	1,2
3	1,2
4	1,2

No	Penelusuran
5	1,2
6	1,2
7	1
8	1
9	1
10	1

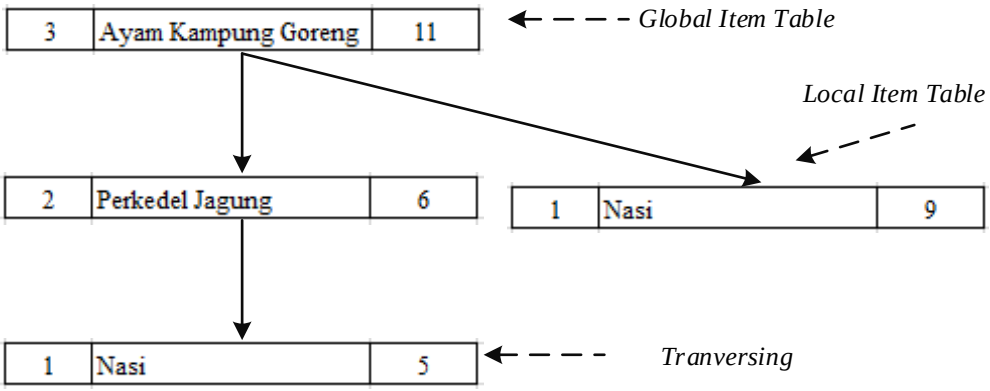
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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* 3:

Index	Nama Menu	*Support Count*	PST
1	Nasi	9	
2	Perkedel Jagung	6	



Gambar 3. 39 *Local CFP-Tree index* 3

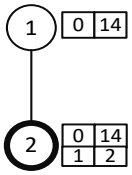
Setelah *Local CFP-Tree index 3* terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 3, dimana *index* 3 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 3. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 3 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.40:



Gambar 3.40 *Local CFP-Tree Projection* 3

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 3 dengan Nama Menu Ayam Kampung Goreng adalah sebagai berikut: (3,2:6), (3,2,1:5), dan (3,1:9).

- Pencarian *Local Frequent Pattern Tree* untuk *index* 2. Cari jalur (node) pada *Global CFP-Tree* yang hanya berakhiran 2, untuk jelas nya dapat dilihat pada gambar 3.41.



Gambar 3. 41 *CFP-Tree* Berakhiran *index* 2

Berikut hasil penelusuran untuk *index* 2 pada *Global CFP-Tree* yang dapat dilihat pada tabel 3.45.

Tabel 3.45 Penelusuran *Index* 2

No	Penelusuran
1	2
2	2
3	2,1
4	2,1
5	2,1

No	Penelusuran
6	2,1
7	2,1
8	2,1
9	2,1
10	2,1
11	2,1
12	2,1
13	2,1
14	2,1
15	2,1
16	2,1

Dari gambar 3.41 diatas didapat jumlah *support count* dari setiap *node* yaitu 1(14) karena *minimum support*-nya adalah 5, maka *node* 1 di masukan pada tabel *local item*. Berikut adalah *local item table* yang terbentuk, diacu pada tabel 3.46.

Tabel 3.46 *Local Item Table index* 2

Index	*Node*	Nama Menu	*Support Count*
1	1	Nasi	14

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 3.47.

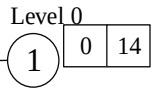
Tabel 3.47 Hasil *Mapping Local CFP-Tree*

No	Penelusuran
1	1
2	1

No	Penelusuran
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1

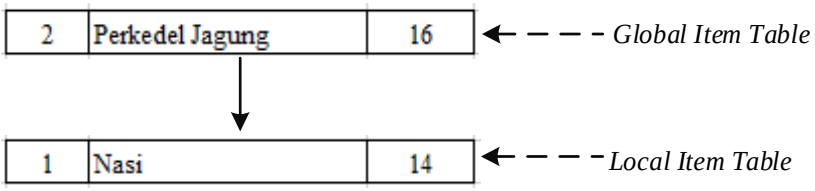
Setelah proses *mapping*, selanjutnya melakukan pembentukan *local CFP-Tree*. Aturan pembentukan *Local CFP-Tree* sama dengan pembentukan *Global CFP-Tree*, yang membedakan adalah pada *Global CFP-Tree* yang digunakan dalam pembentukan *tree*-nya adalah *Global Item table* yang terbentuk dari *Global 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* 2:

Index	Nama Menu	*Support Count*	PST
1	Nasi	14	



Gambar 3.42 *Local CFP-Tree index* 2

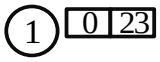
Setelah *Local CFP-Tree index 2* terbentuk langkah selanjutnya adalah membuat *Local CFP-Tree projection* untuk *index* 2, dimana *index* 2 akan menjadi *root* dan setiap *index Local CFP-Tree* akan menjadi anak dari *index* 2. Untuk menemukan *local frequent pattern tree,* lakukan penelusuran di *Local CFP-Tree* di *index* 2 mulai dari *path* sampai *root* secara *battom-up,* untuk lebih jelasnya dapat dlihat pada gambar 3.43:



Gambar 3.43 *Local CFP-Tree Projection* 2

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *index* 2 dengan Nama Menu Perkedel Jagung adalah sebagai berikut: (2,1:14).

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



Gambar 3. 44 *CFP-Tree* Berakhiran *index 1*

Dari gambar 3.44 diatas tidak ada *node* yang ditemukan, maka tidak terbentuk *Local Item Table* sehingga untuk *Local CFP-Tree* dan *Frequent Pattern*nya juga tidak akan terbentuk.

Berikut merupakan hasil *frequent pattern* dari *Local CFP-Tree* dapat dilihat pada tabel 3.48

Tabel 3. 48 Hasil *Frequent Pattern*

Index	*Frequent Pattern*
14	{14,1:5}
13	{13,1:5}
12	{12,1:5}
11	{11,1:5}
10	{10,2:6}
9	{9,1:6}
7	{7,4:5} {7,4,1:5} {7,3:5} {7,3,1:6} {7,2:6} {7,2,1:5} {7,1:10}
6	{6,3:5} {6,3,1:6} {6,2:6} {6,2,1:6} {6,1:10}
5	{5,1:7}
4	{4,2:6} {4,2,1:6} {4,1:11}
3	{3,2:6} {3,2,1:5} {3,1:9}
2	{2,1:14}

Dari hasil *frequent pattren* yang didapat pada tabel 3.48 langkah selanjutnya yaitu mencari *subset* dari masing-masing *index* seperti pada tabel 3.9:

Tabel 3. 49 Pembentukan *Subset*

Index	***Subset***
14	{14},{1},{14,1},{1,14}
13	{13},{1},{13,1},{1,13}
12	{12},{1},{12,1},{1,12}
11	{11},{1},{11,1},{1,11}
10	{10},{2},{10,2},{2,10}
9	{9},{1},{9,1},{1,9}
7	{7},{4},{7,4},{4,7},{7},{4},{1},{7,4,1},{4,7,1},{1,7,4},{7},{3},{7,3},{3,7} {7},{3},{1},{7,3,1},{3,7,1},{1,7,3},{7},{2},{7,2},{2,7},{7},{2},{1},{7,2,1},
	{2,7,1},{1,2,7},{7},{1},{7,1},1,7}
6	{6},{3},{6,3},{3,6},{6,3,1},{3,6,1},{1,3,6},{6},{2},{6,2},{2,6},{6},{2},{1}, {6,2,1},{2,6,1},{1,2,6},{6},{1},{6,1},{1,6}
5	{5},{1},{5,1},{1,5}
4	{4},{2},{4,2},{2,4},{4},{2},{1},{4,2,1},{2,4,1},{1,2,4},{4},{1},{4,1},{1,4}
3	{3},{2},{3,2},{2,3},{3},{2},{1},{3,2,1},{2,3,1},{1,2,3},{3},{1},{3,1},{1,3}
2	{2},{1},{2,1},{1,2}

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

Dari *frequent pattern* yang didapat, akan di-*generate* untuk mendapatkan *rule* dengan *minimum confidence* = 60%. Dihitung dengan persamaan 2.4 sebagai berikut :

$$\text{Confidence} = (A \rightarrow B) = \frac{\textit{Jumlah transaksi yang mengandung item A dan B}}{\textit{Jumlah total Transaksi A}} * 100\%$$

(Persamaan 2-4)

Berikut adalah hasil generate rule diacu pada tabel 3.50 :

Tabel 3. 50 Hasil *Generate Rule*

Rule	***Confidence***
14 → 1	(5/5) * 100% = 100%
13 → 1	(5/5) * 100% = 100%
12 → 1	(5/6) * 100% = 83.3333%
11 → 1	(5/6) * 100% = 83.3333%
10 → 2	(6/6) * 100% = 100%
9 → 1	(6/7) * 100% = 85.7143%
7 → 6	(5/10) * 100% = 50%
7 → 6, 1	(5/10) * 100% = 50%
7 → 4	(6/10) * 100% = 60%
7 → 4, 1	(6/10) * 100% = 60%
7 → 2	(5/10) * 100% = 50%
7 → 2, 1	(5/10) * 100% = 50%
7 → 1	(10/10) * 100% = 100%
6 → 4	(6/10) * 100% = 60%
6 → 4, 1	(6/10) * 100% = 60%
6 → 2	(6/10) * 100% = 60%
6 → 2, 1	(6/10) * 100% = 60%
6 → 1	(10/10) * 100% = 100%
5 → 1	(7/10) * 100% = 70%
4 → 2	(6/11) * 100% = 54.5455%
4 → 2, 1	(6/11) * 100% = 54.5455%
4 → 1	(11/11) * 100% = 100%
3 → 2	(6/11) * 100% = 54.5455%
3 → 2, 1	(5/11) * 100% = 45.4545%
3 → 1	(9/11) * 100% = 81.8182%
2 → 1	(14/16) * 100% = 87.5%

Setelah perhitungan *confidence* diketahui maka selanjutnya dilakukan pengecekan, jika tidak memenuhi *minimum confidence* menu tersebut akan dihapus untuk menemukan *rule*. Setiap *id global item* akan dikembalikan sesuai dengan nama menu dari masing-masing *id* agar pembacaan *rule* lebih jelas. Berikut adalah hasil *rule* yang terbentuk sementara dapat dilihat pada tabel 3.51:

Tabel 3.51 Hasil *rule* yang terbentuk sementara

Rule	**Paket Nasi Kotak**	***Confidence***
14 → 1	Telor Balado → Nasi	100%
13 → 1	Sate Usus → Nasi	100%
12 → 1	Cumi Bumbu → Nasi	83.33%
11 → 1	Gepuk Sapi → Nasi	83.33%
10 → 2	Udang Goreng → Perkedel Jagung	100%
9 → 1	Sop Buntut → Nasi	85.71%
7 → 4	Pepes Tahu → Kentang Mustofa	60%
7 → 4, 1	Pepes Tahu → Kentang Mustofa, Nasi	60%
7 → 1	Pepes Tahu → Nasi	100%
6 → 4	Ayam Balado → Kentang Mustofa	60%
6 → 4, 1	Ayam Balado → Kentang Mustofa, Nasi	60%
6 → 2	Ayam Balado → Perkedel Jagung	60%
6 → 2, 1	Ayam Balado → Perkedel Jagung, Nasi	60%
6 → 1	Ayam Balado → Nasi	100%
5 → 1	Tempe Orek → Nasi	70%
4 → 1	Kentang Mustofa → Nasi	100%
3 → 1	Ayam Kampung Goreng → Nasi	81.82%
2 → 1	Perkedel Jagung → Nasi	87.50%

Setalah mengeliminasi *rule* yang tidak memenuhi *minimum confidance* tahap selanjutnya adalah membentuk paket nasi kotak dalam bentuk awal dengan menjumlahkan setiap harga menu dalam satu *rule*. Berikut merupakan Hasil pembentukan paket nasi kotak dalam bentuk awal dapat dilihat pada tabel 3.52

Tabel 3.52 Paket Nasi Kotak dalam bentuk awal

Paket Nasi Kotak	Total
Telor Balado → Nasi	Rp.8.000
Sate Usus → Nasi	Rp.7.000
Cumi Bumbu → Nasi	Rp.15.000
Gepuk Sapi → Nasi	Rp.19.000
Udang Goreng → Perkedel Jagung	Rp.10.000
Sop Buntut → Nasi	Rp.29.000
Pepes Tahu → Kentang Mustofa	Rp.12.000
Pepes Tahu → Kentang Mustofa, Nasi	Rp.16.000
Pepes Tahu → Nasi	Rp.12.000
Ayam Balado → Kentang Mustofa	Rp.19.000
Ayam Balado → Kentang Mustofa, Nasi	Rp.23.000
Ayam Balado → Perkedel Jagung	Rp.17.000
Ayam Balado → Perkedel Jagung, Nasi	Rp.21.000
Ayam Balado → Nasi	Rp.19.000
Tempe Orek → Nasi	Rp.8.000
Kentang Mustofa → Nasi	Rp.8.000
Ayam Kampung Goreng → Nasi	Rp.20.000
Perkedel Jagung → Nasi	Rp.6.000

Dalam menentuan pembentukan paket nasi kotak yang akan ditawarkan ke konsumen pihak rumah makan memiliki batasan untuk penentuan isi menu dan batasan harga dalam satu paket. Berikut adalah Batasan Paket Nasi Kotak:

1. Dalam 1 paket minimal isi tiga menu makanan termasuk sudah ada nasi
2. Untuk harga minimal Rp.15.000 dan maksimal Rp.30.000

Berikut merupakan hasil paket nasi kotak yang telah di *filter* dapat dilihat pada tabel 3.53.

Tabel 3.53 Hasil Paket yang terbentuk

Paket	Nama Menu	Harga Rekomendasi
Paket 1	Pepes Tahu ,Kentang Mustofa, Nasi	Rp.16.000
Paket 2	Ayam Balado, Perkedel Jagung, Nasi	Rp.21.000
Paket 3	Ayam Balado, Kentang Mustofa, Nasi	Rp.23.000

3.1.3 Analisis Non Fungsional dan Kebutuhan Non Fungsional

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

3.1.3.1 Analisis Pengguna

Pengguna yang akan mengoperasikan aplikasi ini merupakan manager yang bertindak sebagai pengolah bisnis. Analisis pengguna dapat dilihat pada tabel 3.54

Tabel 3.54 Analisis Pengguna

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

1. Analisis spesifikasi perangkat lunak pada sistem yang sedang berjalan di rumah makan Laksana Hati adalah:
   a. Sitem Operasi Windows 7
   b. Microsoft Office 2010
2. Analisis spesifikasi kebutuhan perangkat lunak untuk menjalankan sistem yang akan dibangun adalah:
   a. Sistem Operasi Windows7
   b. MySQL sebagai database server
   c. Microsoft Visual Studio 2010
3. Evaluasi kebutuhan perangkat lunak

Berdasarkan analisis spesifikasi perangkat lunak yang ada di rumah makan Laksana Hati, spesifikasi perangkat lunak yang digunakan masih kurang mendukang. Oleh karena itu mengatasi hal tersebut maka akan di *install* MySQL dan juga Visual Studio pada computer 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 rumah makan Laksana Hati.

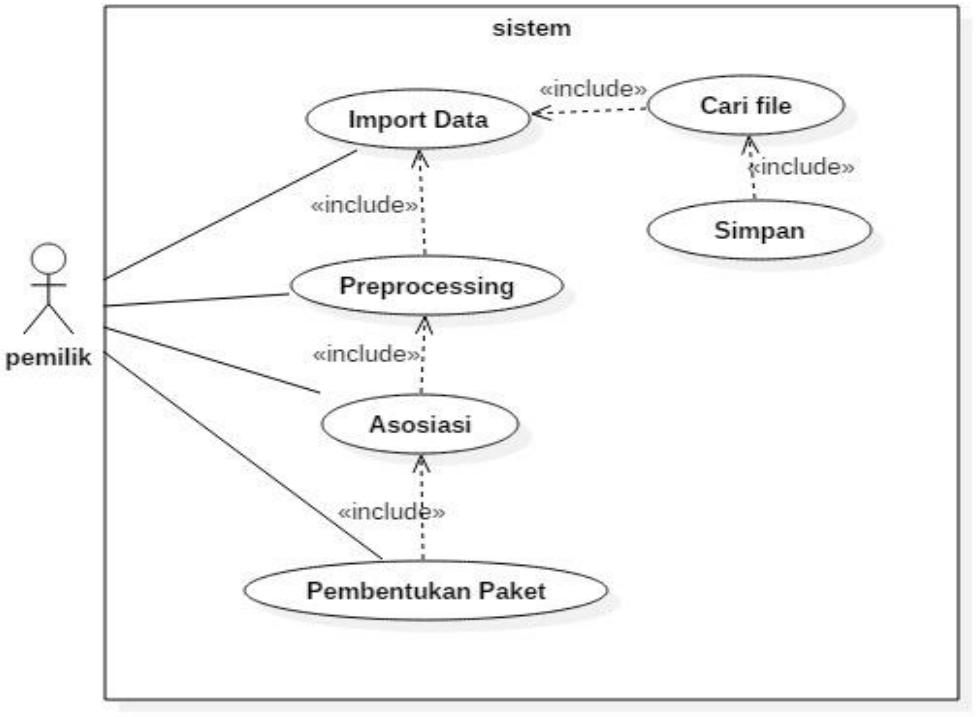
1. Analisis spesifikasi perangkat keras pada sistem yang sedang berjalan di rumah makan Laksana Hati adalah:
   a. *Processor* : Intel Core i3 2333M 2.4Ghz
   b. *Hard disk* : 500 GB
   c. *Memory* : 4 GB
2. Analisis spesifikasi kebutuhan perangkat keras untuk menjalankan sistem yang akan dibangun adalah:
   a. *Processor* : berkecepatan minimal 2 GHz
   b. *Hard disk* : 50 GB
   c. *Memory* : 1 GB
3. Evaluasi kebutuhan perangkat lunak
   Berdasarkan analisis spesifikasi perangkat keras yang ada di rumah makan Laksana Hati, perangkat keras yang digunakan sudah cukup untuk dapat menjalankan aplikasi Data *Mining*.

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah analisis terhadap kebutuhan secara fungsional baik dalam aliran data ataupun informasi. Analisis kebutuhan data fungsional yang digunakan dalam sistem ini di gambarkan dengan analisis pemograman berorientasi objek dengan menggunakan konsep UML (*Unifield Modeling languages*).

3.1.4.1 PDiagram *Use Case*

Use case atau diagram *Use case* merupakan pemodelan kelakuan (behavior) sistem yang akan dibuat. Diagram *use case* dapat dilihat pada gambar 3.45 dan Penjelasan *user* dan *use case* dapat dilihat tabel 3.55 dan table 3.56.



Gambar 3.45 Diagram UseCase

Tabel 3.55 Tabel definisi *user*

User	**Deskripsi**
User	*User* bertugas untuk melakukan *import* data, pilih file, import, proses *preprocessing*, proses asosiasi dan pembentukan paket.

Tabel 3.56 Tabel Deskripsi *Use Case*

No	***Use Case***	**Deskripsi**
1	*Import* Data	*Import* data harus dilakukan *user* terlebih dahulu, data yang di *import* merupakan data transaksi yang

		selanjutnya data tersebut akan disimpan ke dalam *database*.
2	Pilih File	*User* memilih *file* data transaksi
3	*Import*	*User* meng*import* data transaksi ke database
4	*Preprocessing*	Data transaksi yang sudah ter*import* di dalam *database* kemudian akan dilakukan proses pembersihan data, pemilihan atribut dan hitung *support count*
5	Asosiasi	*User* 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	*User* memasukkan nilai parameter yaitu nilai minimal dan maksimal harga, kemudian sistem melakukan pencarian harga yang memenuhi batasan, melakukan filter, menghasilkan informasi paket

3.1.4.1.1 Skenario *Use Case*

1. Skenario *Use Case Import Data*

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

Tabel 3.57 Requirement A.1

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

Tabel 3.58 Skenario Use Case Import Data

Use case Name	***Import Data***	
Related Requirements	Requirement *A.1*	
Goal In Context	*Import* data transaksi Rumah Makan Laksana Hati dan menyimpan ke dalam *database*	
Precondition	*User* menyiapkan data transaksi	
Successful End Condition	Data transaksi yang di *import* oleh *user* berhasil di simpan ke dalam *database*	
Failed End Condition	Data transaksi gagal di simpan ke dalam *database*	
Actors	*User*	
Triger	*User* memasukan data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*User* memilih menu *import* data
	2	Sistem menampilkan *import* data
	3	*User* klik tombol cari *file.*
	4	*User* mencari data transaksi
	5	Sistem meng*generate file* excel
	6	Sistem menampilkan data transaksi ke data grid.
	7	*User* klik tombol simpan
	8	Sistem menyimpan data transaksi ke *database*
	9	Sistem menampilkan pesan data berhasil simpan
Extension	Step	Branching Action
	-	

2. Skenario Pilih *File*

Skenario *use case* pilih *file* menggambarkan langkah–langkah aksi *user* terhadap sistem untuk melakukan pilih *file* data transaksi.

Tabel 3.59 Skenario Use Case Pilih *File*

Use case Name	**Cari *File***	
Related Requirements	Requirement *A.1*	
Goal In Context	Memilih *file* data transaksi rumah makan Laksana Hati yang akan di simpan ke dalam *database*	
Precondition	*User* menyiapkan data transaksi	
Successful End Condition	Data transaksi yang di *import* oleh *user* berhasil di simpan ke dalam *database*	
Failed End Condition	Data transaksi gagal di simpan ke dalam *database*	
Actors	*User*	
Triger	*User* memasukan data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*User* klik tombol cari file
	2	Sistem menampilkan pilih *file*
	3	*User* memilih data transaksi
	4	Sistem melakukan *generate file exel*
	5	Sistem menampilkan data ke data grid
Extension	Step	Branching Action
	-	

3. Skenario Simpan Data

Skenario *use case* simpan data menggambarkan langkah–langkah aksi *user* terhadap sistem untuk melakukan simpan data transaksi.

Tabel 3.60 Skenario Use Case simpan data

Use case Name	**Simpan Data**
Related Requirements	Requirement *A.1*
Goal In Context	Memilih *file* data transaksi rumah makan Laksana Hati yang akan di simpan ke dalam *database*
Precondition	*User* menyiapkan data transaksi

Use case Name	**Simpan Data**	
Successful End Condition	Data transaksi yang di *import* oleh *user* berhasil di simpan ke dalam *database*	
Failed End Condition	Data transaksi gagal di simpan ke dalam *database*	
Actors	*User*	
Triger	*User* memasukan data transaksi	
Included Cases	-	
Main Flow	Step	Action
	1	*User* klik tombol simpan
	2	Sistem menyimpan data transaksi ke *database*
	3	Sistem menampilkan pesan data berhasil disimpan
Extension	Step	Branching Action
	-	

4. Skenario *Use Case Preprocesing*

Skenario *use case preprocessing* menjelaskan langkah-langkah untuk melakukan proses Pembersihan Data, Pemilihan Atribut, dan Hitung *support count*

Tabel 3.61 Requirement A.2

***Requirement* A.2**
Sistem menyediakan menu *preprocesing* untuk melakukan proses *preprocessing*

Tabel 3. 62 Skenario *Use Case Preprocesing*

Use case Name	***Preprocesing***
Related Requirements	Requirement *A.1,* Requirement *A.2*
Goal In Context	Sistem melakukan proses pembersihan data, pemilihan atribut, dan *support count*
Precondition	Data transaksi sudah ter*import* didalam *database*
Successful End Condition	Sistem melakukan proses pembersihan data, pemilihan atribut, dan *support count*

Use case Name	***Preprocesing***	
Failed End Condition	Sistem gagal untuk melakukan proses pembersihan data, pemilihan atribut, dan *support count*	
Actors	*User*	
Triger	*User* meminta proses pembersihan data, pemilihan atribut, dan *support count*	
Included Cases	Pembersihan Data, Pemilihan Atribut, dan Hitung support count	
Main Flow	Step	Action
	1	*User* memilih menu *preprocessing*
	2	Sistem menampilkan menu *preprocessing*
	3	*User* klik tombol proses *preprocessing*
	4	Sistem melakukan proses pembersihan data yang berkategori jenis minuman dan item tunggal
	5	Sistem menyimpan hasil pembersihan data ke *database*
	6	Sistem menampilkan hasil pembersihan data
	7	Sistem melakukan proses seleksi atribut
	8	Sistem menyimpan hasil seleksi atribut ke dalam database
	8	Sistem menampilkan hasil seleksi Atribut
	9	sistem melakukan proses Hitung *support count* tiap menu
	10	Sistem menyimpan hasil proses Hitung Hitung support count kedalam *database*
	11	Sistem menampilkan hasil proses Hitung support count

Main Flow	***Step***	***Action***
	12	Sistem menampilkan pesan *preprocessing* berhasil.
Extension	**Step**	**Branching Action**
	-	

5. Skenario Use Case Proses Asosiasi

Skenario *use case* proses asosiasi menggambarkan langkah-langkah aksi *user* untuk melakukan proses memasukan nilai minimum *support, confidence,* lihat *generate rules* dan lihat r*ules*. Skenario *use case* proses asosiasi dapat dilihat pada table 3.64

Tabel 3.63 Requirement A.3

***Requirement* A.3**
Sistem menyediakan menu asosiasi untuk memasukan nilai minimum *support, confidence,* lihat *generate rules* dan lihat r*ules*. Skenario *use case* proses asosiasi dapat dilihat pada tabel 3.64

Tabel 3. 64 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	*User*
Triger	*User* memasukan nilai *Minimum Support* dan *Minimum Confidence*

Use case Name	***Asosiasi***	
Included Cases	Memasukan nilai minimum *support confidence*, lihat hasil *generate rules*, Lihat *Rules*.	
Main Flow	**Step**	**Action**
	1	*User* memilih menu asosiasi
	2	Sistem menampilkan menu asosiasi
	3	*User* memasukan nilai minimum *suppot* dan *confidence*
	4	*User* klik tombol proses
	5	Sistem melakukan proses seleksi sesuai dengan nilai minimum *support*
	6	Sistem mendapatkan *frequent itemset*
		Sistem melakukan proses pencarian *frequent itemset* sesuai nilai minimum *support*
		Sistem menampilkan hasil menampilkan hasil *generate rule*
Extension	**Step**	**Branching Action**
	-	

6. Skenario *Use Case* Pembentukan Paket

Skenario *use case* pembentukan paket menggambarkan langkah-langkah aksi *user* untuk melakukan proses memasukan harga dan pembentukan paket. Skenario *use case* pembentukan Paket dapat dilihat pada table 3.65

Tabel 3.65 *Requirement* A.4

***Requirement* A.4**
Sistem menyediakan menu pembentukan paket untuk memasukan berupa minimal Harga dan maksimal untuk membentuk menu apa saja yang muncul sesuai harga.

Tabel 3.66 Skenario *Use Case* Pembentukan Paket

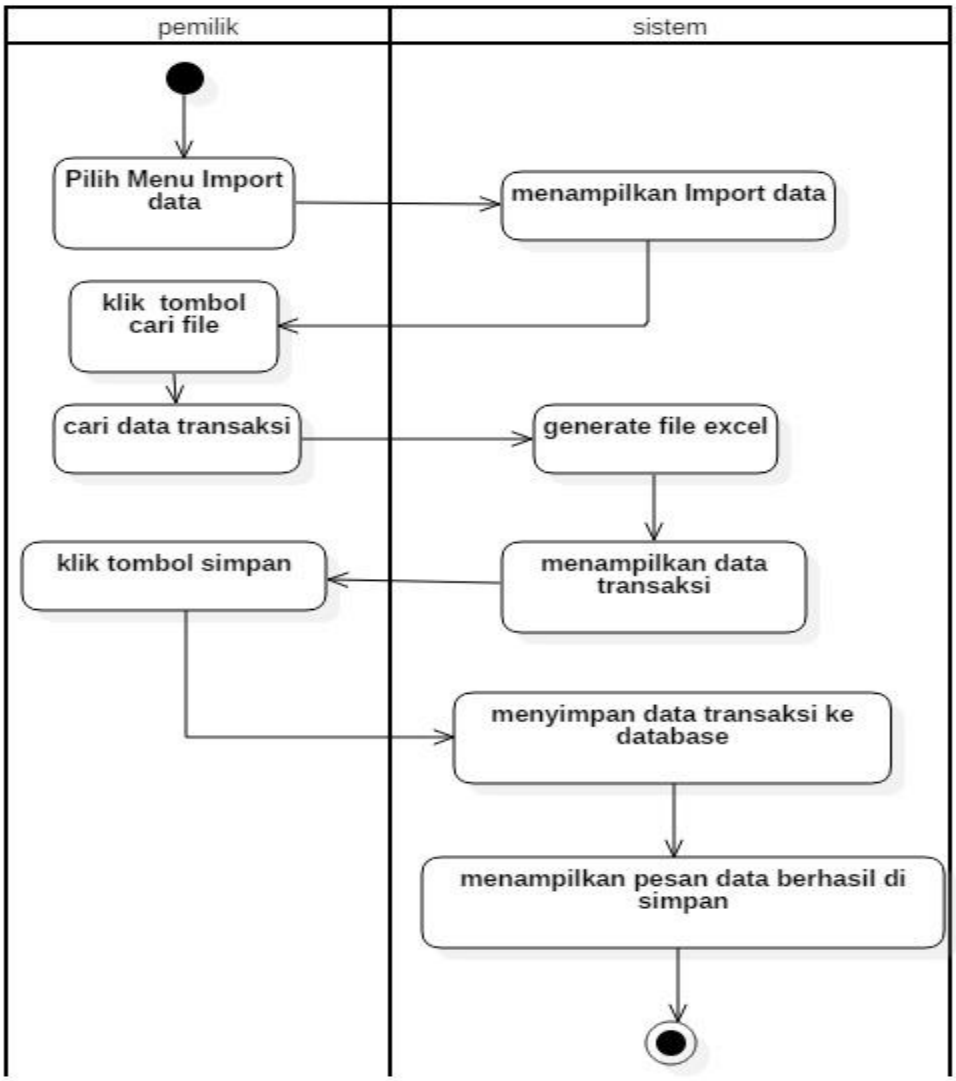
Use case Name	**Asosiasi**	
Related Requirements	Requirement *A.1*, Requirement *A.2*, Requirement *A.3*, Requirement *A.3*	
Goal In Context	Sistem mendapatkan hasil filter *rule* yang memenuhi nilai minimum support dan confidence yang dimasukan	
Precondition	Data sudah melalui proses *Asosiasi*	
Successful End Condition	Sistem berhasil filter *rule* yang memenuhi nilai minimum support dan confidence yang dimasukan	
Failed End Condition	Sistem gagal memfilter *rule* yang memenuhi nilai parameter yang dimasukan	
Actors	*User*	
Triger	*User* memasukan berupa nilai minimum harga dan maksimum harga	
Included Cases	Memasukan harga, Lihat Pembentukan Paket	
Main Flow	**Step**	**Action**
	1	*User* memilih menu pembentukan paket
	2	Sistem menampilkan menu pembentukan paket
	3	*User* memasukan nilai minimum harga dan maksimum harga
	4	Sistem menghitung jumlah harga dari setiap *rule*
	5	Sistem memfilter *rule* yang memenuhi nilai minimum harga dan maksimum harga
	6	Sistem menampilkan data paket yang memenuhi batasan harga
Extension	**Step**	**Branching Action**
	-	

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

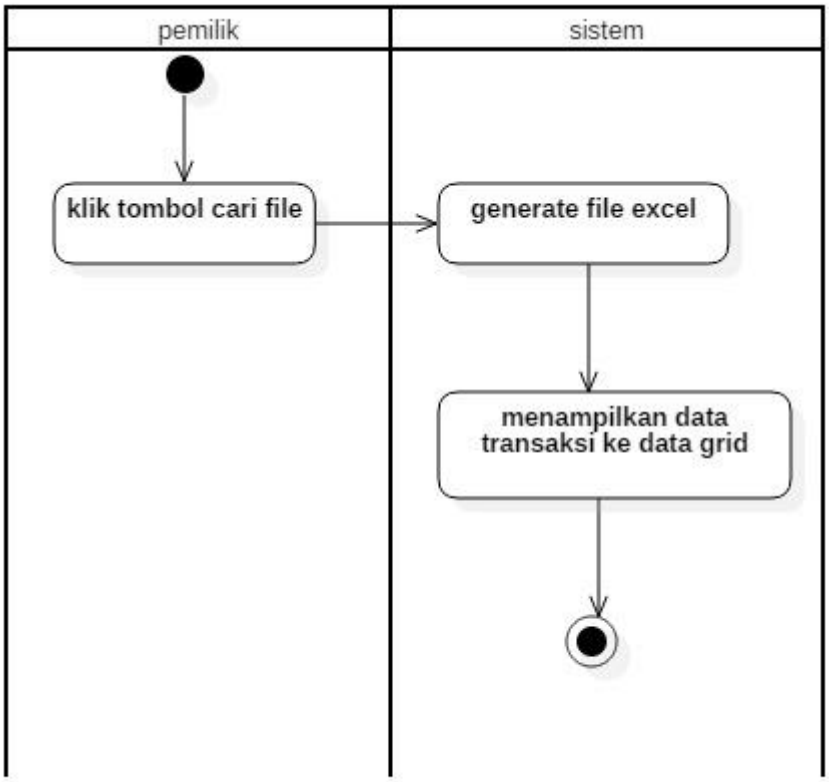
   Activity diagram import data menjelaskan aktivitas aktor dalam melakukan proses import data. Penjelasan activity *diagram import* data diacu pada Gambar 3.46.



Gambar 3. 46 *Activity Diagram* Import Data

2. *Activity Diagram* pilih *File*

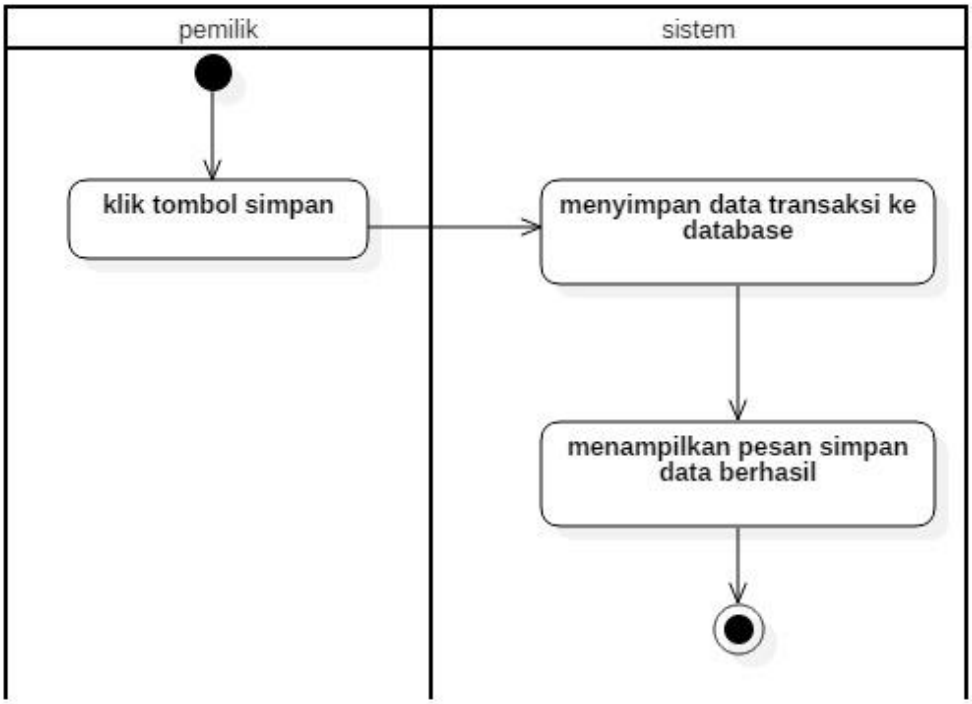
   Activity Diagram pilih *File* menjelaskan aktivitas sistem dalam melakukan proses pemilihan *File. Activity Diagram* pilih *File* dapat dilihat pada gambar 3.47



Gambar 3. 47 *Activity Diagram* Cari *File*

3. *Activity Diagram* Simpan

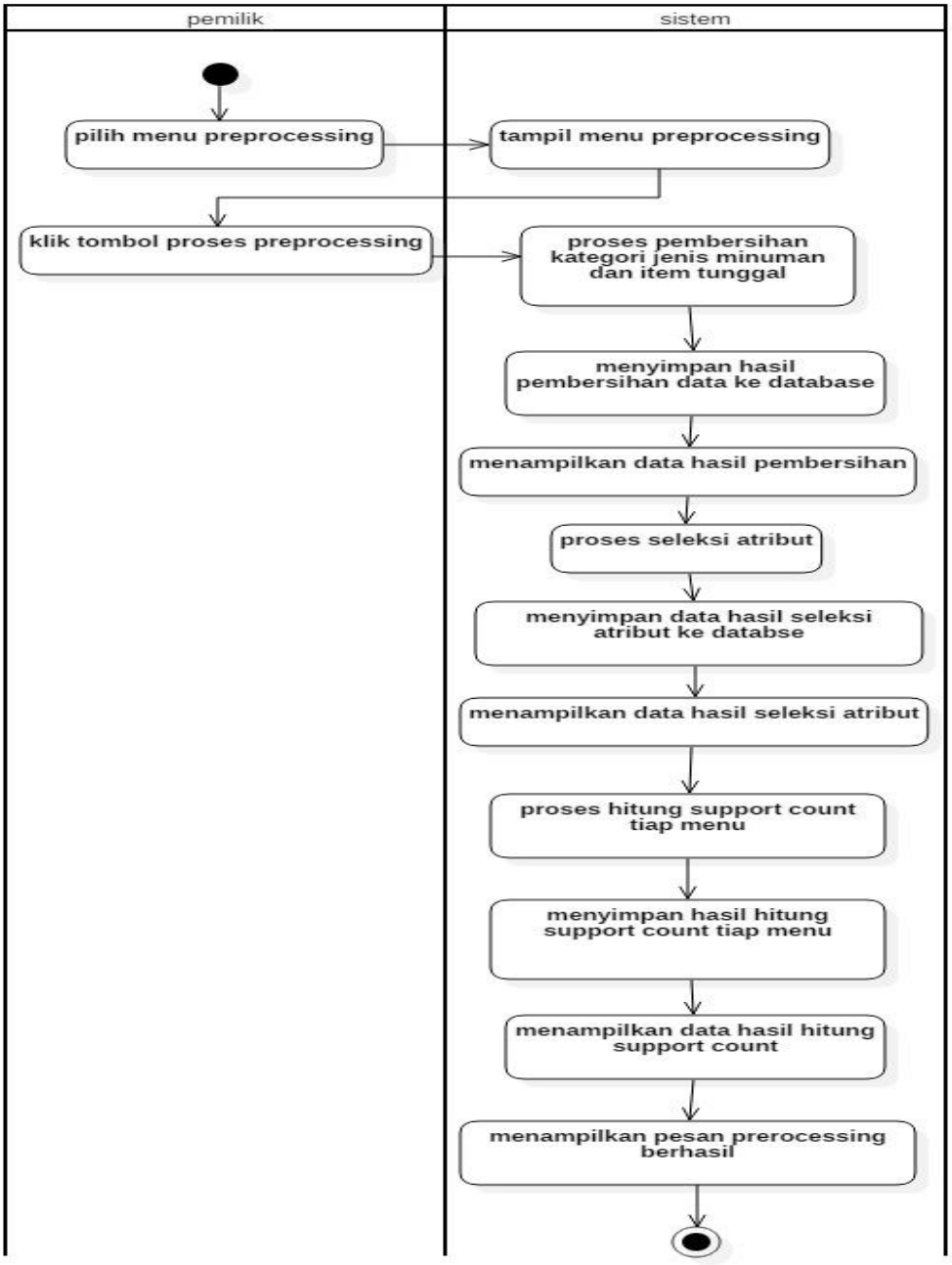
Activity Diagram Simpan menjelaskan aktivitas sistem dalam melakukan proses *import* ke *database. Activity Diagram* Simpan dapat dilihat pada gambar 3.48



Gambar 3. 48 *Activity Diagram* Simpan

4. *Activity Diagram* Preprocesing

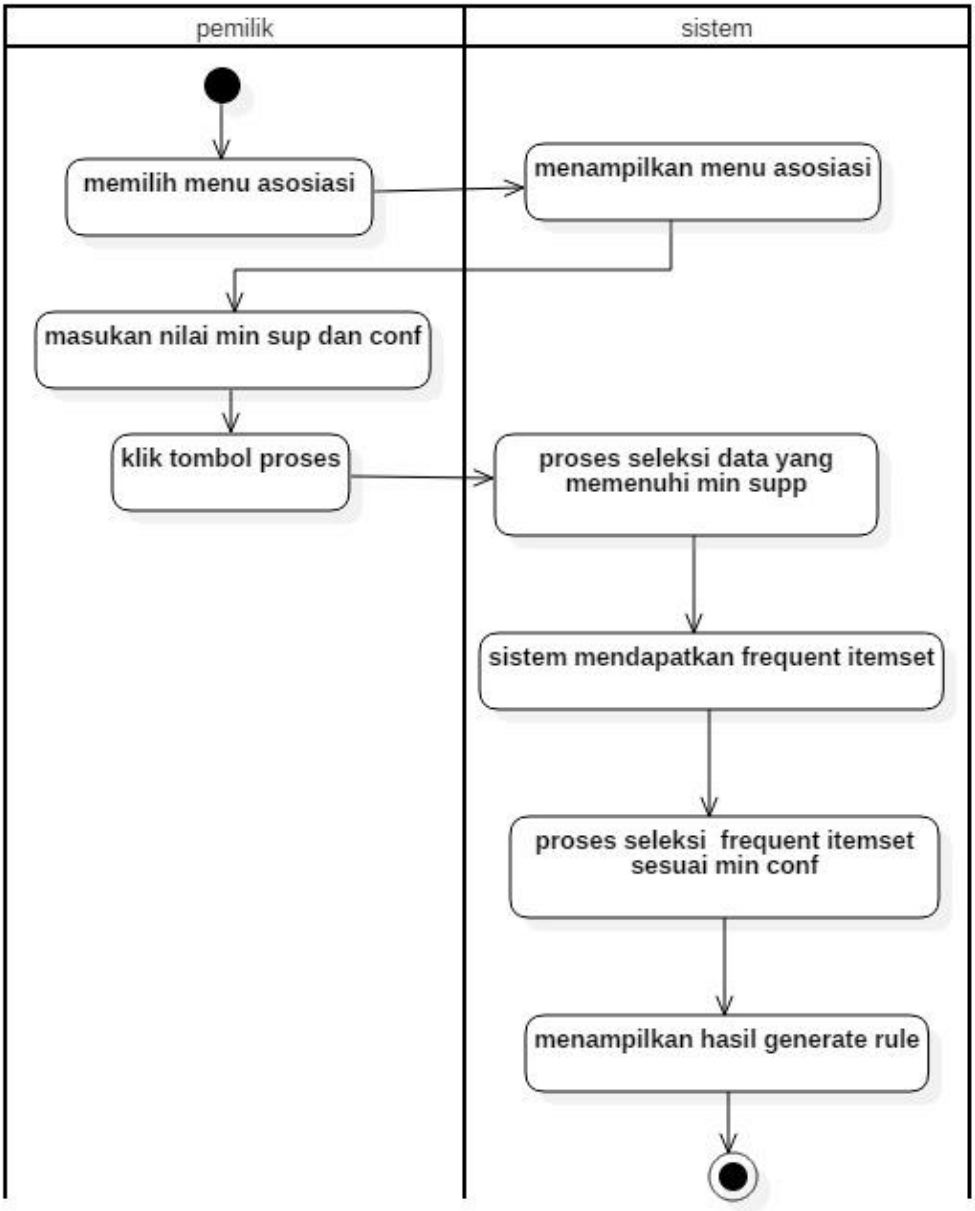
Activity diagram preprocessing menjelaskan aktivitas sistem dalam melakukan proses Pembersihan Data, Pemilihan Atribut dan Hitung *support* count. *Activity diagram* preprocessing dapat dilihat pada gambar 3.49.



Gambar 3. 49 *Activity Diagram* Preprocessing

5. *Activity Diagram* Asosiasi

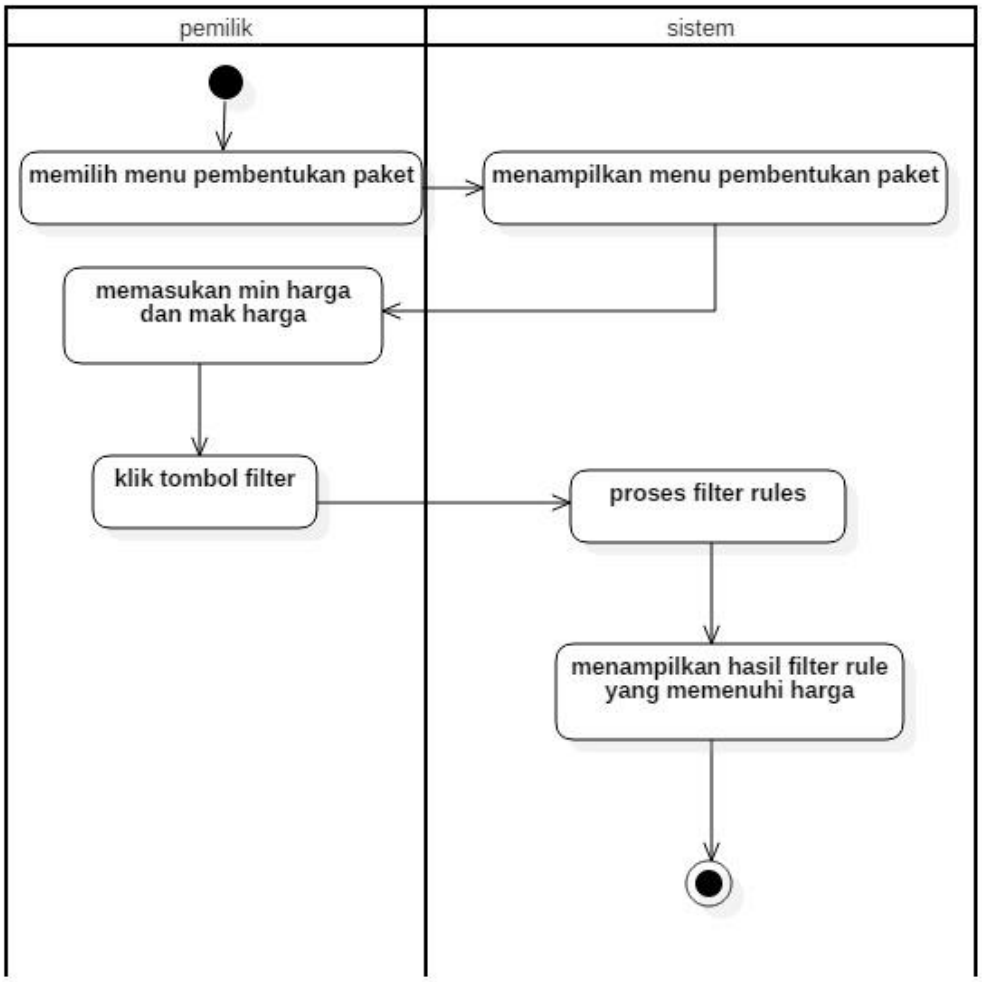
Activity diagram asosiasi menjelaskan aktivitas user dan sistem dalam melakukan proses memasukan nilai *support*, *confidence* lihat hasil *generate rules* dan melihat rule. *Activity diagram* Asosiasi dapat dilihat pada gambar 3.50.



Gambar3.50 *Activity Diagram* Asosiasi

6. *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 dan melihat hasil pembentukan paket. *Activity diagram* lihat hasil pembentukan paket dapat dilihat pada gambar 3.51.



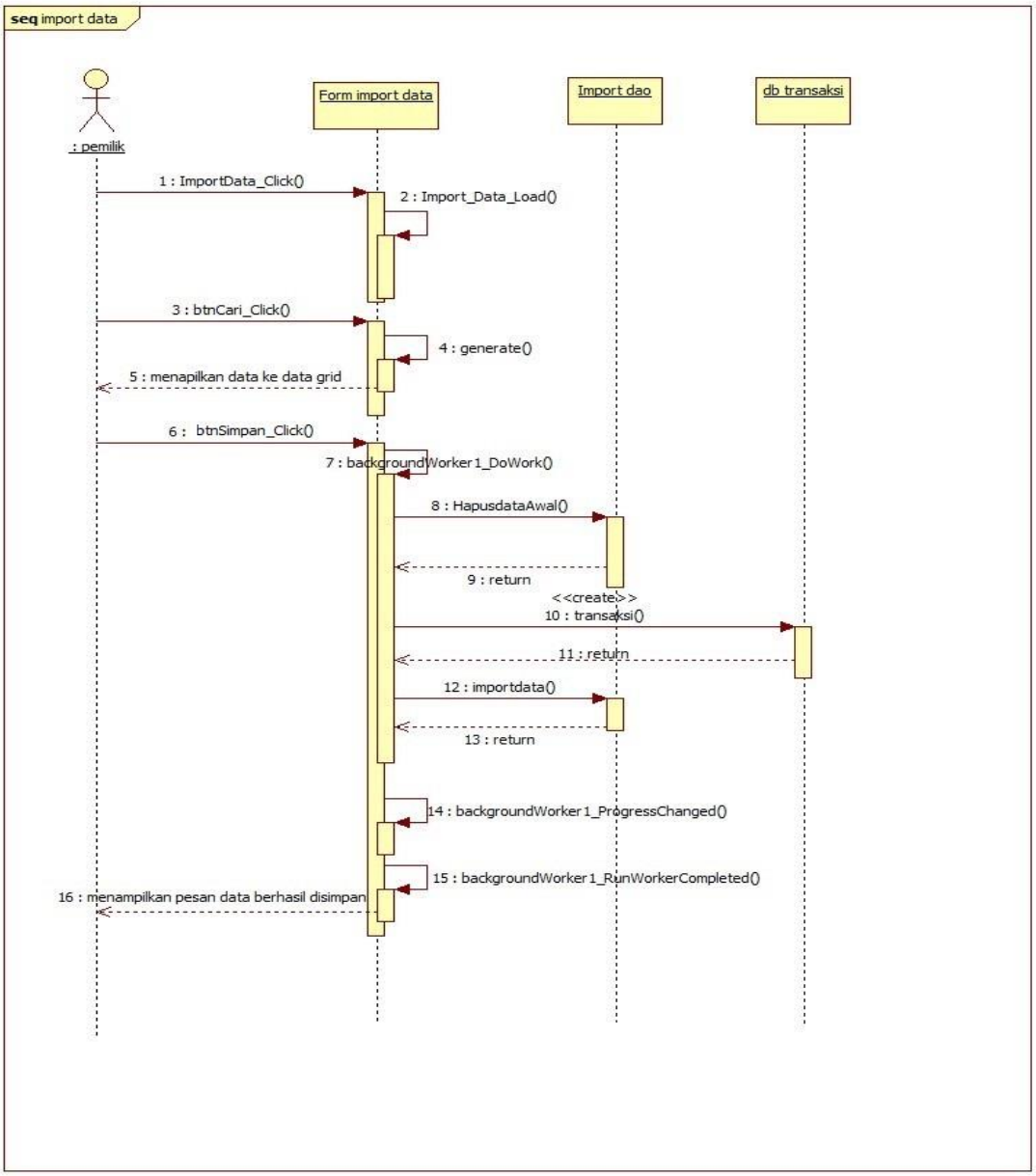
Gambar 3. 51 *Activity Diagram* Pembentukan Paket

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. *Sequens* Diagram Proses *Import Data*

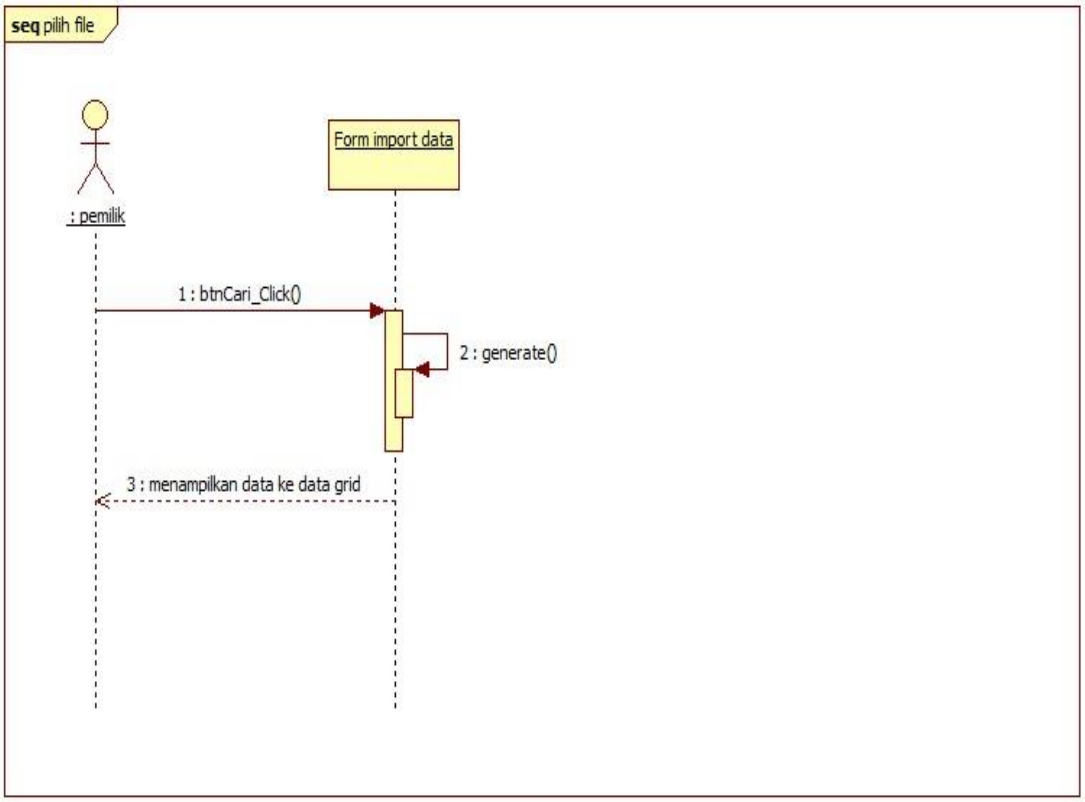
   Untuk *sequence diagram* pada proses import data dapat dilihat pada gambar 3.52.



Gambar 3. 52 *Squence Diagram Import Data*

2. *Sequence Diagram* Proses Cari File :

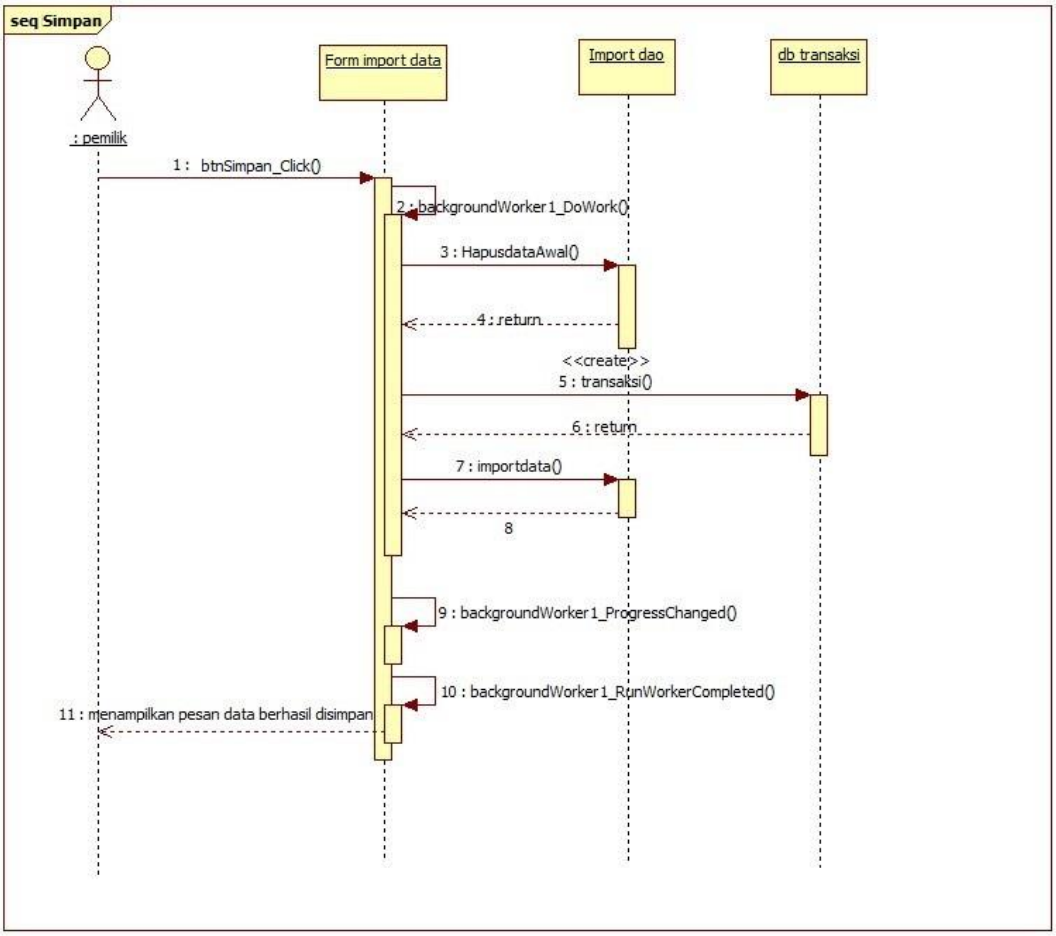
Untuk *sequence diagram* pada proses Cari File data dapat dilihat pada gambar 3.53.



Gambar 3. 53 *Squence Diagram* Cari File

3. *Sequence Diagram* Proses Simpan:

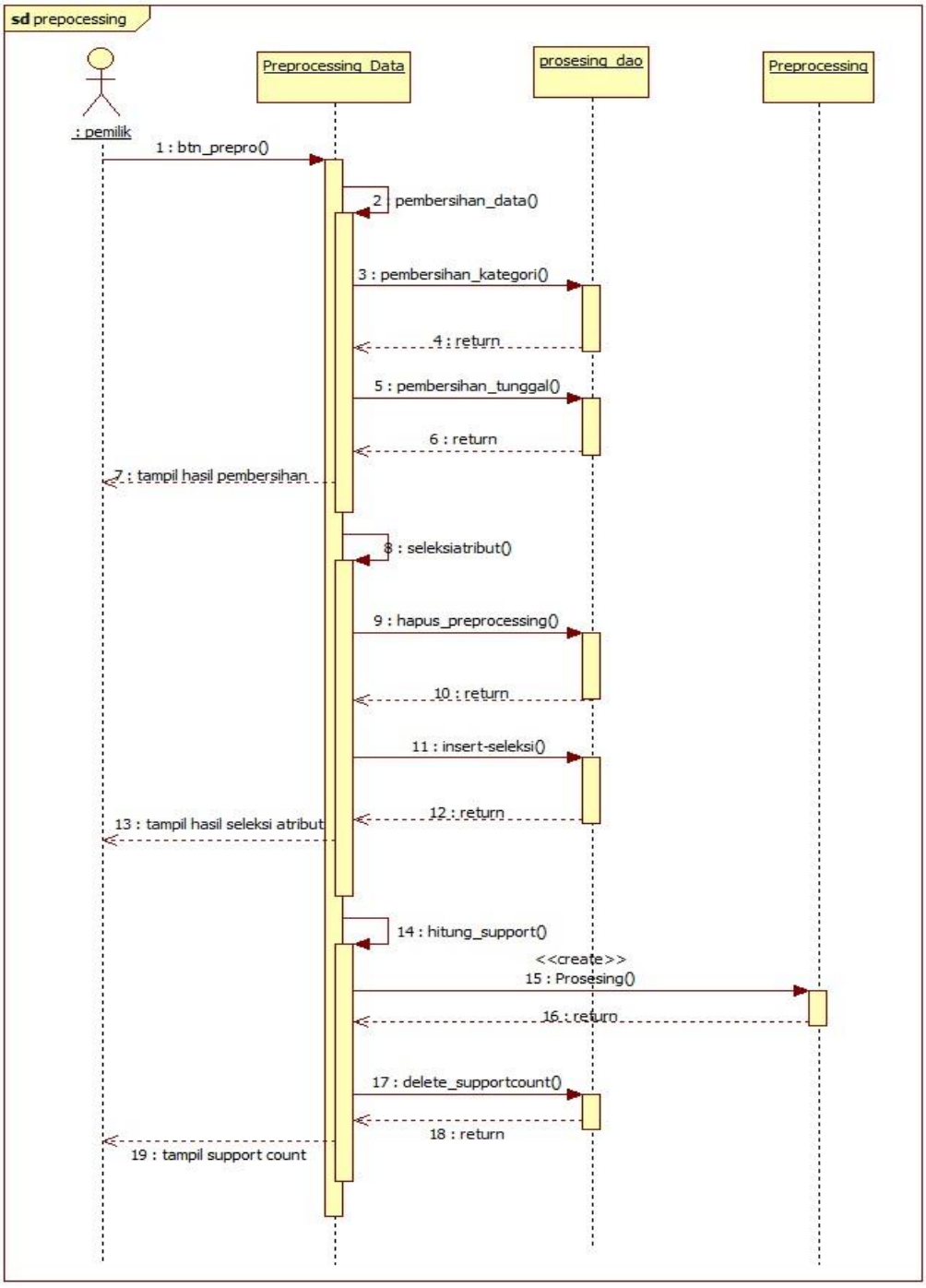
Untuk *sequence diagram* pada proses Simpan data dapat dilihat pada gambar 3.54.



Gambar 3.54 *Squence Diagram* Simpan

4. *Squence Diagram* Proses *Preprocessing*

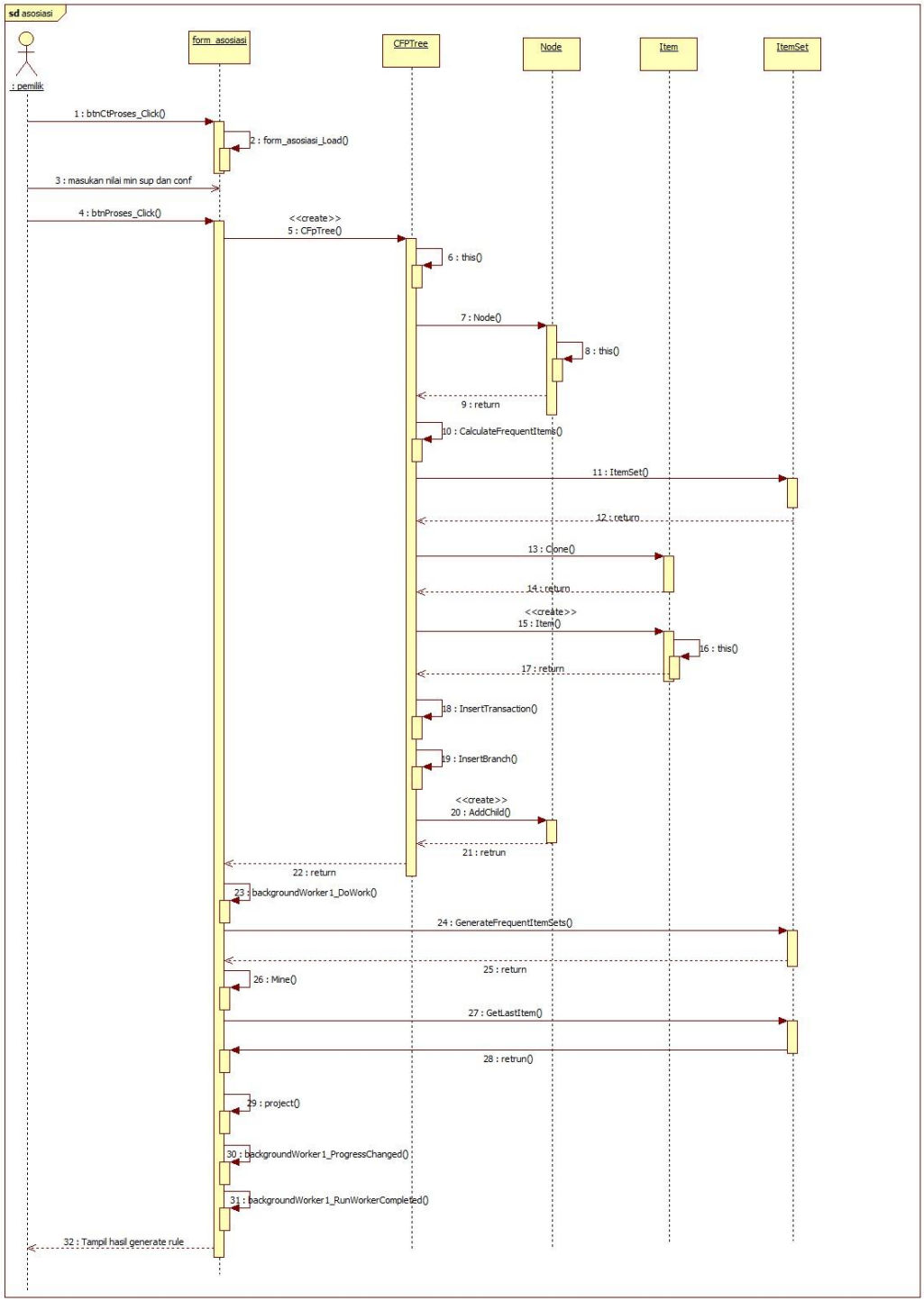
Untuk *sequence diagram* pada proses *Preprocessing* dapat dilihat pada gambar 3.55.



Gambar 3. 55 *Squence Diagram* ***Preprocessing***

5. *Squence Diagram* Proses Asosiasi

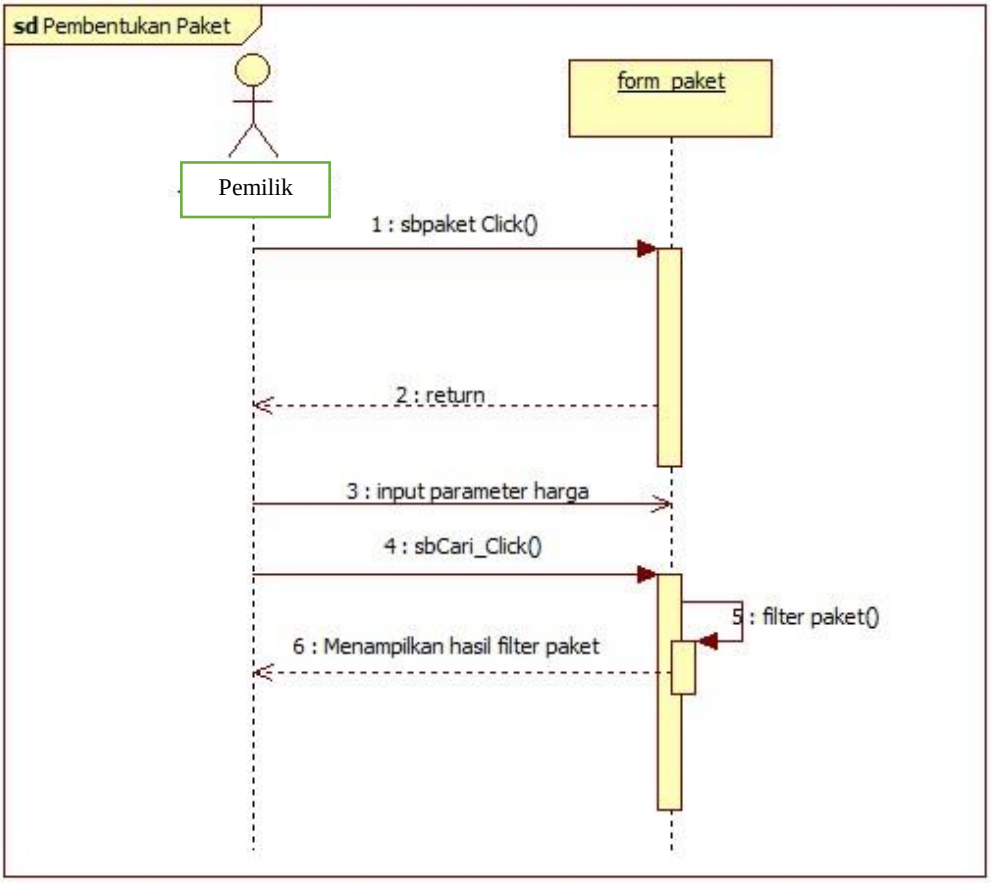
Untuk *sequence diagram* pada proses asosiasi dapat dilihat pada gambar 3.56



Gambar 3. 56 *Squence Diagram* Proses Asosiasi

6. *Squence Diagram* Proses Pembentukan Paket

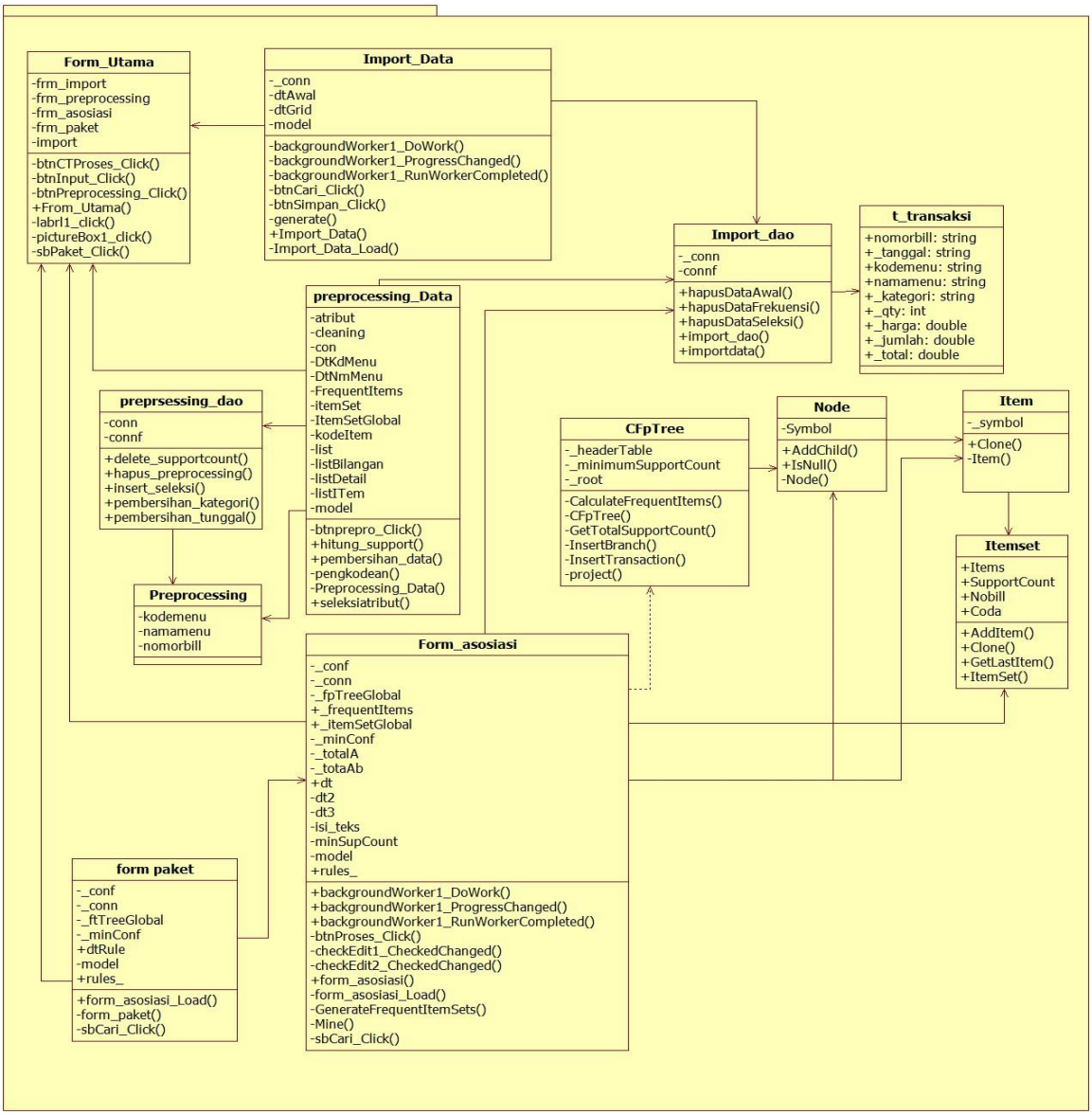
Untuk *sequence diagram* pada Proses Pembentukan Paket dapat dilihat pada gambar 3.57.



Gambar 3.57 *Squence Diagram* Pembentukan Paket

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 3.58.



Gambar 3. 58 *Class Diagram*

3.2 Perancangan Sistem

Proses perancangan yang akan dilakukan pada tahap ini adalah perancangan sistem basis data, perancangan struktur menu, perancangan antar muka, perancangan pesan, jaringan semantik, dan perancangan prosedural.

1.2.1 Perancangan Kelas

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

Tabel 3.67. Struktur Kelas Form_Utama

Class : Form_Utama		
Nama	Jenis	Type
frm_import	Atribut	Private
frm_preprocessing	Atribut	Private
frm_asosiasi	Atribut	Private
frm_paket	Atribut	Private
import	Atribut	Private
btnCTProses_Click()	Method	Private
btnInput_Click()	Method	Private
sbPreprocessing_Click()	Method	Private
From_Utama()	Method	Public
labrl1_click()	Method	Private
pictureBox1_click()	Method	Private
sbPaket_Click()	Method	Private

Tabel 3.68. Struktur Kelas Import_data

Class : Import_data		
Nama	Jenis	Type
_conn	Atribut	Private
dtAwal	Atribut	Private
dtGrid	Atribut	Private
model	Atribut	Private
Import_data()	Method	Public

btnPilih_Click()	Method	Private
generate()	Method	Private
simpan_data()	Method	Private
backgroundWorker1_DoWork()	Method	Private
backgroundWorker1_ProgressChanged()	Method	Private
backgroundWorker1_RunWorkerCompleted()	Method	Private
Import_Data_Load()	Method	Private
Simpan_Click()	Method	Private

Tabel 3.69. Struktur Kelas Import_dao

Class : Import_dao		
Nama	Jenis	Type
_conn	Atribut	Private
connf	Atribut	Private
import_dao()	Method	Public
hapusDataAwal()	Method	Public
hapusDataFrekuensi()	Method	Public
hapusDataJoin()	Method	Public
hapusDataSeleksi()	Method	Public
Importdata()	Method	Public

Tabel 3.70. Struktur Kelas preprocessing

Class : preprocessing		
Nama	Jenis	Type
atribut	Atribut	Private
cleaning	Atribut	Private
con	Atribut	Private
DtKdMenu	Atribut	Private
DtNmMenu	Atribut	Private
FrequentItems	Atribut	Private

itemSet	Atribut	Private
ItemSetGlobal	Atribut	Private
kodeItem	Atribut	Private
list	Atribut	Private
listBilangan	Atribut	Private
listDetail	Atribut	Private
listITem	Atribut	Private
model	Atribut	Private
btnAtribut_Click()	Method	Private
btnCleaning_Click()	Method	Private
btnFrequency_Click()	Method	Private
sortingData()	Method	Private

Tabel 3.71. Struktur Kelas Form_asosiasi

Class : form_asosiasi		
Nama	Jenis	Type
_conf	Atribut	Private
fpTreeGlobal: FpTree	Atribut	Private
_minConf: double	Atribut	Private
_conf: double	Atribut	Private
_frequentItems: List	Atribut	Public
_itemSetGlobal: List	Atribut	Public
minSupCount: int	Atribut	Private
model	Atribut	Private
rules_	Atribut	Private
totalAb: int	Atribut	Public
totaAb: int	Atribut	Public
form_asosiasi()	Method	Public
form_asosiasi_Load()	Method	Private
btnProses_Click()	Method	Private
GenerateFrequentItemSets()	Method	Private

Mine()	Method	Private
backgroundWorker1_DoWork()	Method	Private
backgroundWorker1_ProgressChanged()	Method	Private
backgroundWorker1_RunWorkerCompleted()	Method	Private
sbCari_Click()	Method	Private

Tabel 3.72. Struktur Kelas Form_paket

Class : form_paket		
Nama	Jenis	Type
_conf	Atribut	Private
_conn	Atribut	Private
FrequentItems: List	Atribut	Private
ItemSetGlobal: List	Atribut	Private
dtRule	Atribut	Public
model	Atribut	Private
rules_	Atribut	Public
form_asosiasi_Load()	Method	Private
form_paket()	Method	Private
sbCari_Click()	Method	Private

Tabel 3.73. Struktur Kelas CFpTree

Class : CFpTree		
Nama	Jenis	Type
_headerTable	Atribut	Private
_minimumSupportCount	Atribut	Private
_root	Atribut	Private
CalculateFrequentItems()	Method	Private
CFpTree()	Method	Private
GetTotalSupportCount()	Method	Private
InsertBranch()	Method	Private

InsertTransaction()	Method	Private
project()	Method	Private

Tabel 3.74. Struktur Kelas Node

Class : Node		
Nama	Jenis	Type
symbol	Atribut	Private
IsNull()	Method	Public
AddChild()	Method	Public
Node()	Method	Private

Tabel 3.75. Struktur Kelas Item

Class : item		
Nama	Jenis	Type
symbol:	Atribut	Private
Item()	Method	Public
Clone()	Method	Private

Tabel 3.76. Struktur Kelas Itemset

Class : Itemset		
Nama	Jenis	Type
Items	Atribut	Public
SupportCount	Atribut	Public
Nobill	Atribut	Public
Coda	Atribut	Public
ItemSet()	Method	Public
AddItem()	Method	Public
GetLastItem()	Method	Public
Clone()	Method	Public

Tabel 3.77. Struktur t_transaksi

Class : t_transaksi		
Nama	Jenis	Type
nomorbill: string	Atribut	Public
_tanggal: string	Atribut	Public
kodemenu: string	Atribut	Public
namamenu: string	Atribut	Public
_kategori: string	Atribut	Public
_qty: int	Atribut	Public
_harga: double	Atribut	Public
_jumlah: double	Atribut	Public
_total: double	Atribut	Public

Tabel 3.8. Struktur preprosessing_dao

Class : preprosessing_dao		
Nama	Jenis	Type
conn	Atribut	private
connf	Atribut	private
delete_supportcount()	Method	Public
hapus_preprocessing()	Method	Public
insert_seleksi()	Method	Public
pembersihan_kategori()	Method	Public
pembersihan_tunggal()	Method	Public

1.2.2 Perancangan Data

1.2.2.1 Perancangan Sistem Basis Data

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

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

1.2.2.2 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 3.78 sampai tabel 3.81.

1. Tabel Transaksi

Tabel transaksi merupakan tabel yang digunakan untuk menyimpan data transaksi hasil import dari file excel dapat dilihat pada tabel 3.78.

Tabel 3. 78 Data Awal Transaksi

Nama *Field*	Tipe Data	Panjang	Keterangan
id	Int	11	*Primari Key*
nomor_bill	Varchar	20	
tanggal	Varchar	20	
kode menu	Varchar	6	
nama_menu	Varchar	50	
kategori	Varchar	20	
qty	Int	10	
Harga	Varchar	30	
subtotal	Varchar	30	
total	Varchar	30	

2. Tabel *Preprocessing*

Tabel *preprocessing* adalah tabel yang digunakan untuk menyimpan data hasil *preprocessing* dapat dilihat pada tabel 3.79.

Tabel 3. 79 Data *Preprocessing*

Nama Field	Tipe Data	Panjang	Keterangan
nomor_bill	Varchar	20	
kode_menu	Varchar	20	
nama_menu	Varchar	50	

3. Tabel *Support Count*

Tabel *Support Count* adalah tabel yang digunakan untuk menyimpan data hasil penghitungan kemunculuan banyaknya masing-masing jenis menu dapat dilihat pada tabel 3.80.

Tabel 3. 80 Data *Support Count*

Nama Field	Tipe Data	Panjang	Keterangan
kode_menu	Varchar	20	
frekuensi	int	11	

4. Tabel *Rules*

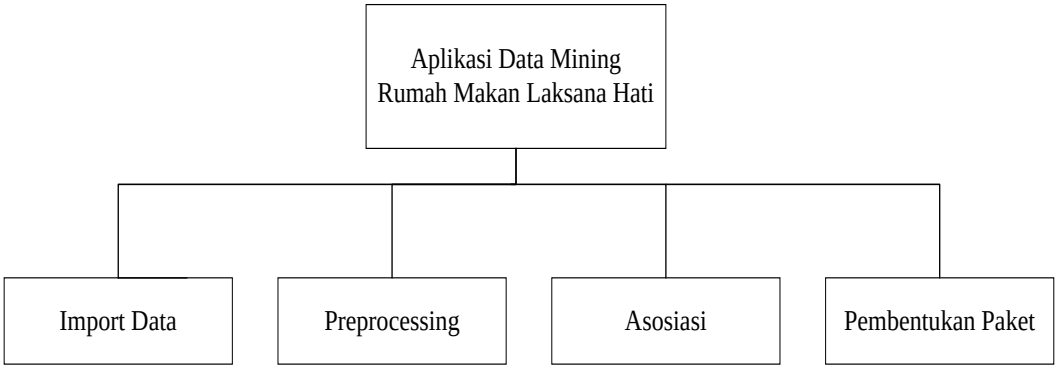
Tabel rules adalah tabel yang digunakan untuk menyimpan hasil pencarian *Frequent Itemset* dapat dilihat padatabel 3.81.

Tabel 3. 81 Data Pembentukan *Rules*

Nama Field	Tipe Data	Panjang	Keterangan
rule	Var	100	
support	Int	11	
confidence	float		

1.2.3 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 3.59.



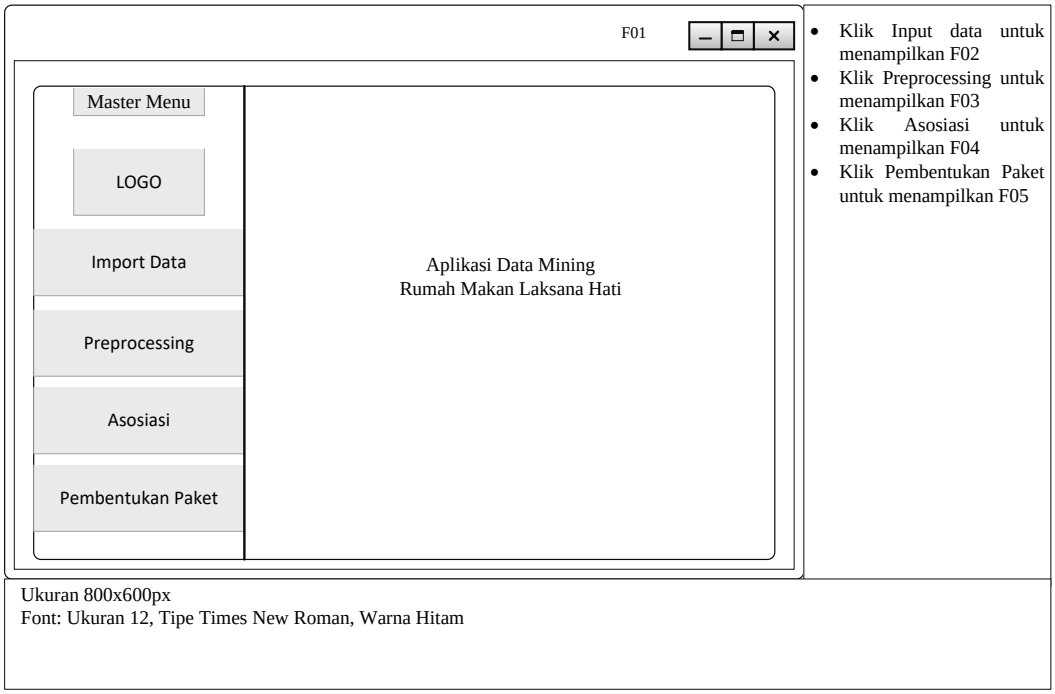
Gambar 3. 59 Struktur Menu

1.2.4 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* rumah makan Laksana Hati adalah sebagai berikut:

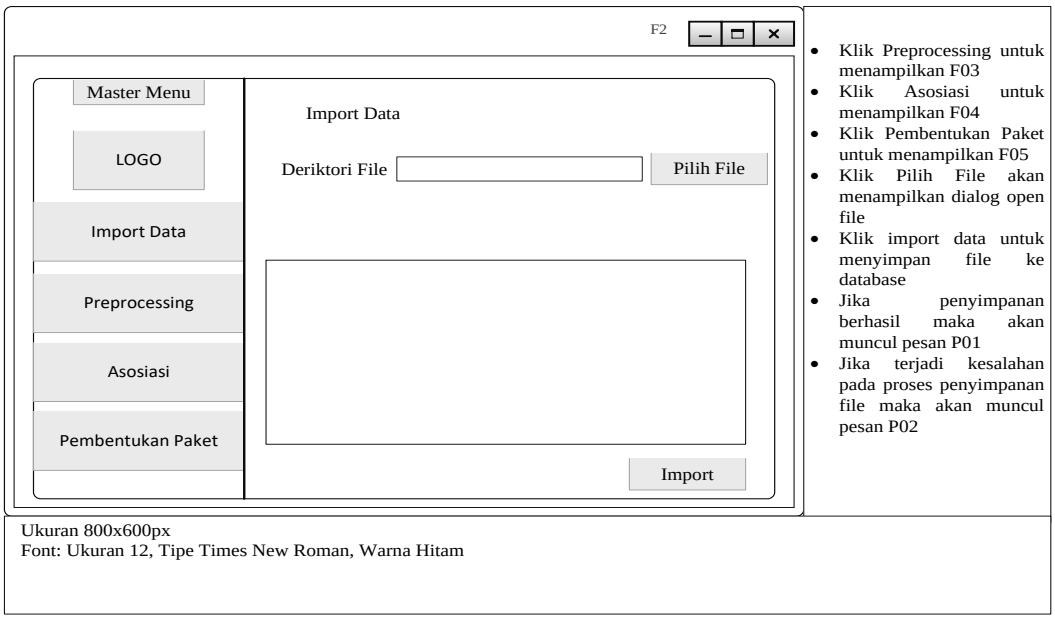
1. Perancangan Antar Muka Menu Utama

   Perancangan Antar Muka Menu Utama dapat dilihat pada gambar 3.60.



Gambar 3. 60 Antar Muka Menu Utama

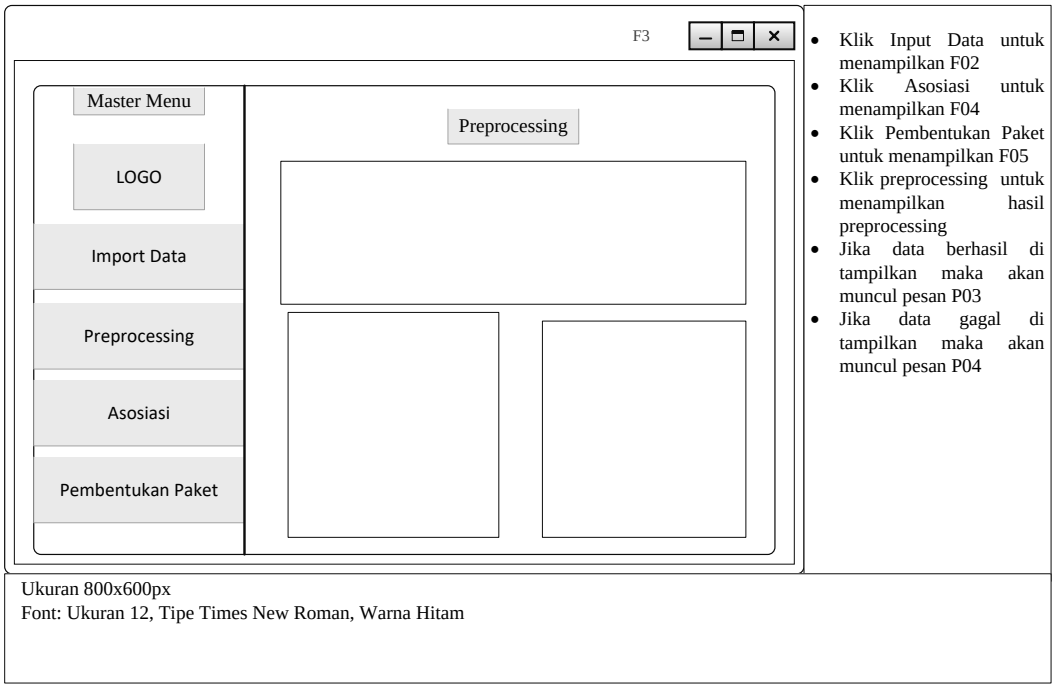
2. Perancangan Antar Muka Import Data Perancangan Antar Muka Input Data dapat dilihat pada gambar 3.61



Gambar 3. 61 Antar Muka Menu Import Data

3. Perancangan Antar Muka Preprocessing Data

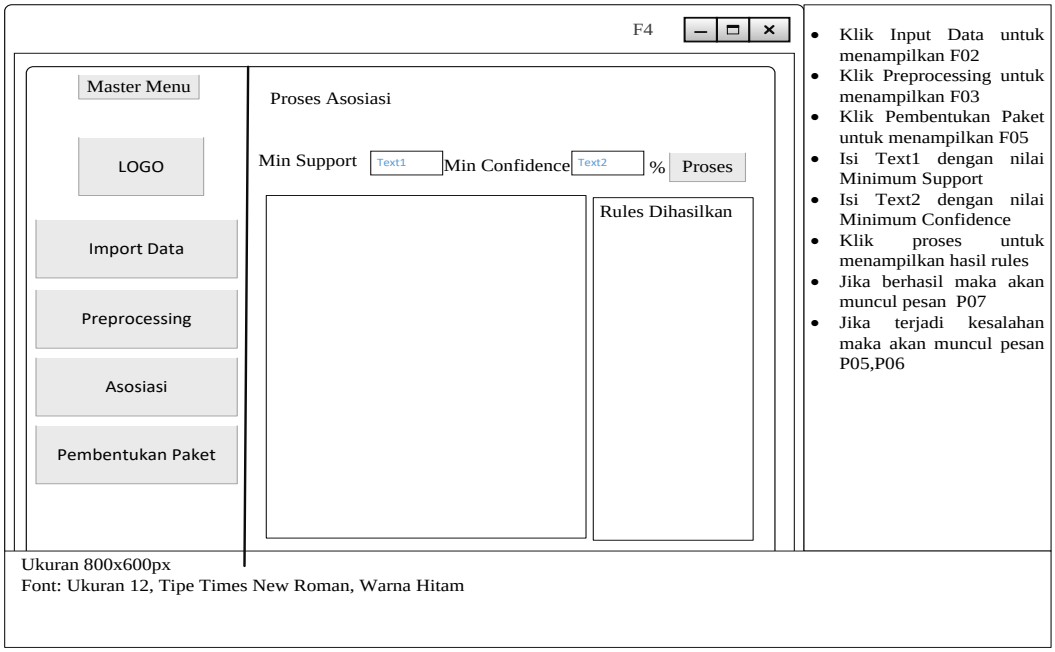
Perancangan Antar Muka Preprocessing Data dapat dilihat pada gambar 3.62



Gambar 3. 62 Antar Muka Preprocessing

4. Perancangan Antar Muka Proses Asosiasi

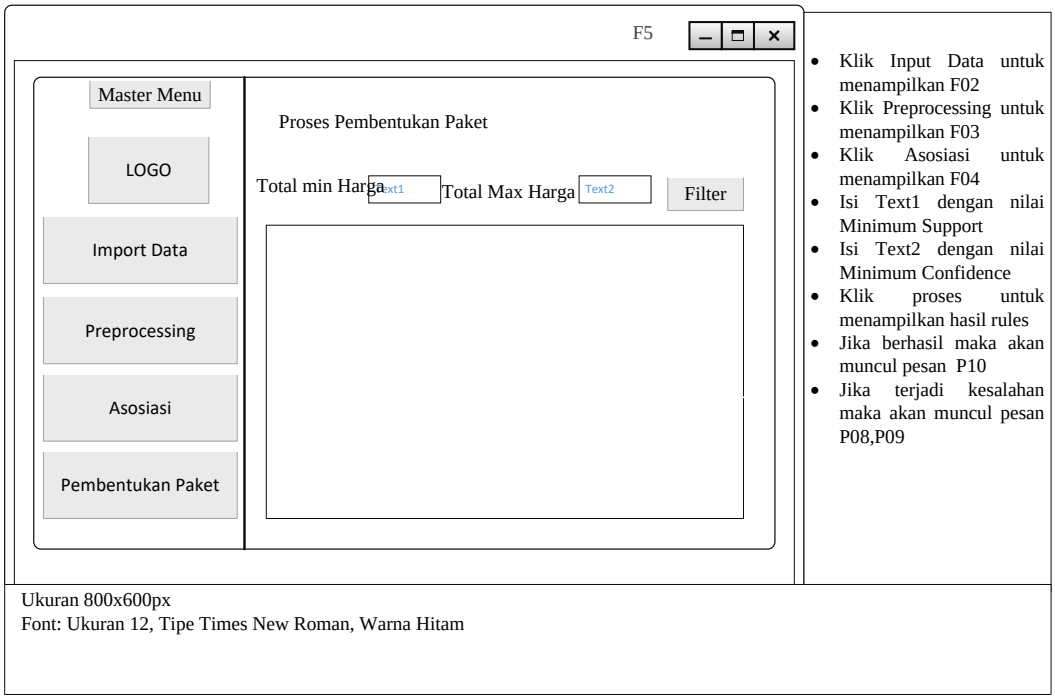
Perancangan Antar Muka Proses Asosiasi dapat dilihat pada gambar 3.63.



Gambar 3. 63 Antar Muka Proses Asosiasi

5. Perancangan Antar Muka Proses Pembentukan Paket

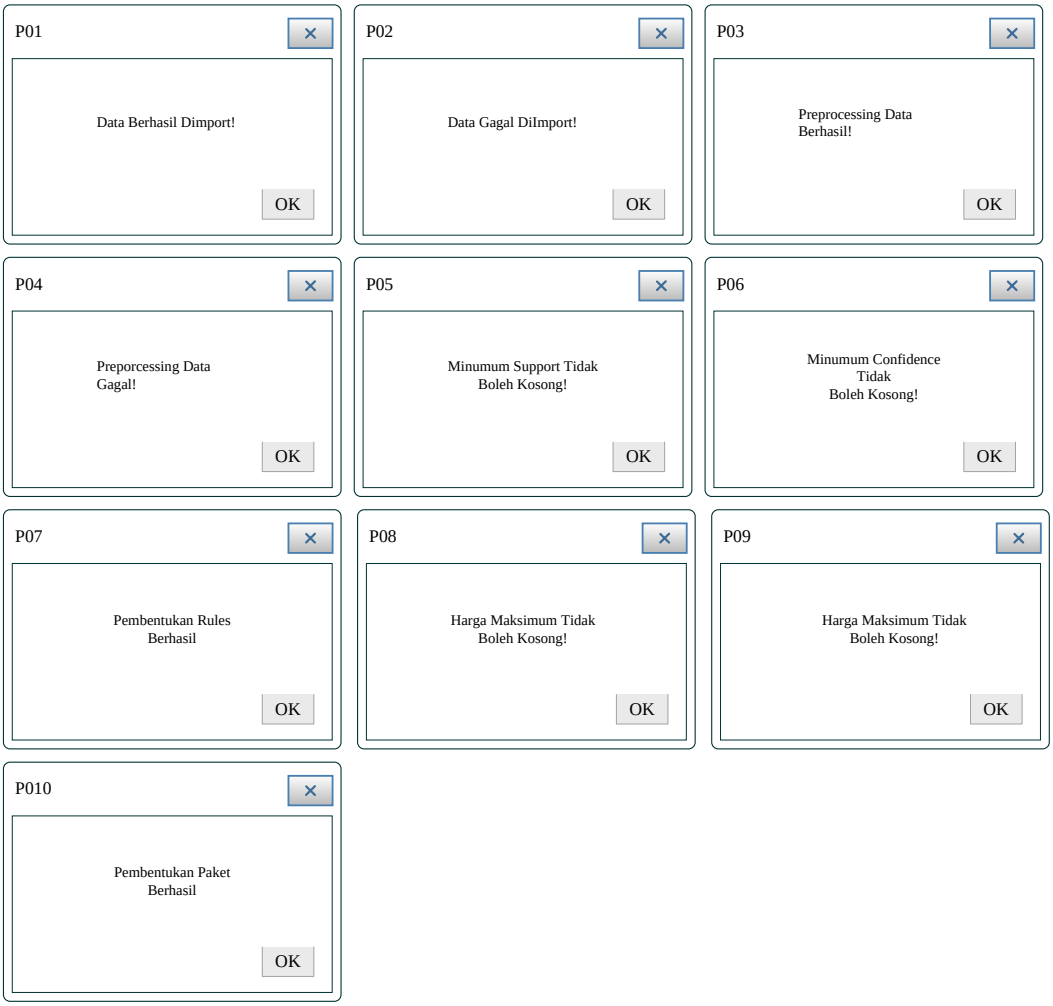
Perancangan Antar Muka Proses Pembentukan Paket dapat dilihat pada gambar 3.64.



Gambar 3. 64 Antar Muka Proses Pembentukan Paket

3.2.5 Perancangan Pesan

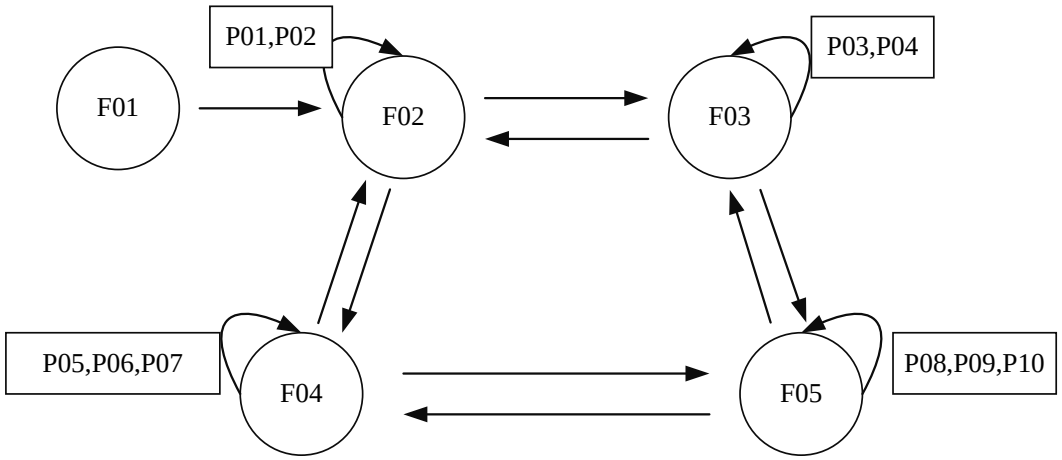
Adapun perancangan pesan dapat dilihat pada gambar 3.65 berikut ini.



Gambar 3. 65 Perancangan Pesan

3.2.6 Perancangan Semantik

Perancangan Semantik dapat dilihat pada gambar 3.66.



Gambar 3. 66 Perancangan Semantik