

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem ialah uraian dari suatu informasi menjadi bagian-bagian dari komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan dan hambatan yang terjadi juga kebutuhan yang diperlukan sehingga dapat diusulkan perbaikan-perbaikannya. Analisis Sistem ini bertujuan untuk menjelaskan permasalahan yang muncul pada saat pembangunan sistem. Dalam analisis sistem ini meliputi beberapa bagian, yaitu:

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

3.1.1. Analisis Masalah

Berdasarkan hasil pengamatan, diketahui bahwa permasalahan yang ada di kafe Kopi Cilik yaitu , banyaknya saingan diluar sana membuat penjualan kafe cenderung menurun dalam beberapa bulan terakhir ini, sehingga muncul pemikiran *manager* untuk menerapkan strategi baru dalam penjualan kafe. *Manager* ingin membuat suatu promosi berupa pemaketan produk dalam penjualannya, karena dengan penjualan berupa paket akan memberikan pilihan baru pada pelanggan dengan harga yang lebih terjangkau. Dengan begitu, penjualan kafe akan lebih maksimal dalam upaya menarik minat konsumen. Untuk pembuatan paket yang akurat, diperlukan pencarian pola pembelian pelanggan pada data transaksi penjualan. Untuk pencarian pola pembelian tersebut maka diperlukan penerapan *data mining* pada data transaksi penjualan dengan metode *association rule*.

3.1.2. Analisis Penerapan Metode CRISP-DM

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

3.1.2.1. *Business Understanding*

Tahapan *business understanding* atau pemahaman bisnis ini berfokus pada pemahaman secara menyeluruh tujuan yang ingin dicapai oleh *client* dari perspektif bisnis [6]. Kemudian pemahaman tersebut diubah menjadi sebuah rencana awal *data mining* yang dirancang untuk mencapai tujuan [6].

1. *Determine Business Objectives*

Tujuan bisnis kafe Kopi Cilik ialah melayani konsumen dengan memasarkan produk yang ada untuk memenuhi permintaan konsumen dengan berbagai macam strategi dan inovasi sehingga konsumen merasa terpuaskan. Salah satu strategi pemasaran yang ingin diterapkan yaitu dengan memanfaatkan penjualan paket. Dengan diterapkannya *data mining* pada data penjualan kafe, dapat menghasilkan pengetahuan berupa keterkaitan antar menu yang sering dibeli oleh konsumen yang dapat dijadikan landasan untuk menentukan menu apa saja yang dapat dijadikan kombinasi pada paket yang akan ditawarkan.

2. *Assess Situation*

Tahap menilai situasi bertujuan untuk menjelaskan fakta-fakta sumber daya, kendala dan batasan yang tersedia di kafe Kopi Cilik [6]. Berikut ini merupakan sumber daya yang ada pada kafe Kopi Cilik dan batasan-batasannya :

a. Sumber Daya

1) Sumber Daya Perangkat Keras

- a. Terdapat Alat penunjang transaksi yaitu printer *thermal* dan *cashdrawer*.
- b. Komputer
- c. Printer

2) Sumber daya data

- a. Sumber daya data yang ada tersimpan di *database* kafe Kopi Cilik.
- b. Data yang digunakan dalam penelitian ini adalah data penjualan bulan Oktober 2018.

- c. Data penjualan yang digunakan merupakan hasil *export* dari aplikasi *Point Of Sale* (POS) yang ada di kafe Kopi Cilik dengan format MS Excel (xlsx).

b. Batasan

1. Hasil dari *data mining* pada penelitian ini berupa *rule* sebagai penentuan menu yang dapat dijadikan satu paket.
2. Hasil dari *generate rule* akan dilakukan *filtering rule*, yaitu menyeleksi beberapa *rule* yang tidak sesuai dengan batasan kriteria bisnis kafe Kopi Cilik dalam membentuk paket.
3. Jika atribut dan nilai pada data tidak lengkap, maka data tersebut akan dianggap *noise* dan dibersihkan pada proses *preprocessing*.
4. Batasan untuk pembentukan paket berupa penentuan rentang harga yang dapat ditentukan oleh *Manager*.

3. *Determine Data Mining Goals*

Tujuan dilakukannya *data mining* ini adalah untuk mencari pola pembelian konsumen pada data sebagai landasan awal untuk menentukan menu apa saja yang dapat dijadikan satu paket yang bisa ditawarkan pada konsumen.

3.1.2.2. *Data Understanding*

Tahap pemahaman data dilakukan dengan mengumpulkan data awal yang diperlukan dari data penjualan di kafe Kopi Cilik, mengidentifikasi masalah kualitas data, dan menemukan wawasan pada data sebagai hipotesa untuk menemukan informasi pada data [6].

1. *Collect Initial Data*

Sumber data yang didapat dalam penelitian ini merupakan data transaksi penjualan yang terjadi di Kafe Kopi Cilik periode Oktober 2018. Berikut ini pada gambar 3.1 merupakan data transaksi penjualan yang terjadi pada bulan Oktober 2018 hasil *export* dari sistem *Point Of Sale* yang ada di kafe Kopi Cilik dalam bentuk excel (.xlsx) :

Tanggal	Bill	No_Meja	Kode_Menu	Nama_Menu	Kategori	Harga	Qty	Total	Total_Bayar
10/1/2018	181000001	4	K0005	Espresso	Kopi	9000	2	18000	18000
10/1/2018	181000002	9	K1007	Mie Goreng Gila	Makanan	10000	1	10000	15000
10/1/2018	181000002	9	K2010	Ice Tea	Minuman	5000	1	5000	15000
10/1/2018	181000003	8	K0003	Cappuccino	Kopi	14000	2	28000	48000
10/1/2018	181000003	8	K3004	Pisang Keju	Cemilan	10000	2	20000	48000
10/1/2018	181000004	15	K1002	Geprek Hejo	Makanan	16500	3	49500	225000
10/1/2018	181000004	15	K1007	Mie Goreng Gila	Makanan	10000	4	40000	225000
10/1/2018	181000004	15	K1012	Geprek Mozzarella	Makanan	18500	3	55500	225000
10/1/2018	181000004	15	K2010	Ice Tea	Minuman	5000	7	35000	225000
10/1/2018	181000004	15	K2001	Thai Lychee Tea	Minuman	15000	3	45000	225000
10/1/2018	181000005	2	K1008	Nasi Goreng Spesial	Makanan	17000	1	17000	49000
10/1/2018	181000005	2	K1010	Katsudon	Makanan	15000	1	15000	49000
10/1/2018	181000005	2	K2003	Milky Tiramisu	Minuman	12000	1	12000	49000
10/1/2018	181000005	2	K2010	Ice Tea	Minuman	5000	1	5000	49000
10/1/2018	181000006	10	K0003	Cappuccino	Kopi	14000	1	14000	41000
10/1/2018	181000006	10	K0006	Americano	Kopi	10000	1	10000	41000
10/1/2018	181000006	10	K3005	Roti Bakar	Cemilan	8500	2	17000	41000
10/1/2018	181000007	6	K1003	Ayam Bakar	Makanan	16000	2	32000	42000
10/1/2018	181000007	6	K2010	Ice Tea	Minuman	5000	2	10000	42000
10/1/2018	181000008	5	K1004	Hot Crispy Chicken	Makanan	15000	1	15000	57000
10/1/2018	181000008	5	K1003	Ayam Bakar	Makanan	16000	1	16000	63000

Gambar 1.1 Data Penjualan Oktober 2018

2. Describe Data

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

Tabel 1.1 Detail tabel transaksi

Keterangan		
Deskripsi	Data ini merupakan hasil <i>export</i> dari <i>database</i> data transaksi penjualan di kafe Kopi Cilik	
Atribut	Tanggal	Tanggal pembelian
	Bill	Nomor pembelian
	No_Meja	Nomor meja
	Kode_Menu	Kode dari setiap menu yang dibeli
	Nama_Menu	Nama dari menu yang dibeli
	Kategori	Kategori dari menu yang dibeli
	Harga	Harga dari menu yang dibeli
	Qty	Jumlah dari menu yang dibeli
	Total	Total harga dari menu yang dibeli
	Total_Bayar	Total harga yang harus dibayar konsumen

3. *Verify Data Quality*

Verifikasi kualitas data merupakan tahap di mana dilakukan pengecekan kelengkapan data, dan apakah data tersebut memiliki *noise* atau tidak [6]. Pada data yang telah dikumpulkan, terdapat fakta-fakta mengenai data tersebut yaitu:

- a. Data pada setiap atribut lengkap (tidak ada atribut ataupun nilai yang hilang).
- b. Jika ada atribut ataupun nilai yang hilang maka akan dianggap *noise*.
- c. Pada penelitian ini data transaksi penjualan yang hanya memiliki 1 *item* dalam sekali pembelian dianggap *noise*, karena penelitian ini mencari asosiasi antar *item*, sedangkan data yang hanya mempunyai satu *item* dalam satu transaksi tidak mempunyai keterkaitan dengan *item* lainnya.

Untuk menghilangkan *noise* pada data transaksi penjualan akan dilakukan proses pembersihan data pada tahap selanjutnya dalam tahapan *data preparation*.

3.1.2.2. *Data Preparation*

Data Preparation ini merupakan tahap dimana akan dilakukan pemilihan tabel dan *field* yang akan digunakan dalam proses *mining* [6]. Persiapan data dilakukan dengan sebutan *Preprocessing*. Tidak semua data atau atribut data dalam data digunakan dalam proses *data mining*, karena data yang digunakan harus disesuaikan dengan kebutuhan. Adapun tahapan-tahapan *preprocessing* data dalam penelitian ini adalah sebagai berikut:

1. *Select Data*

Proses pertama dalam persiapan adalah proses pemilihan data, atribut pada data akan dipilih dan diproses sesuai dengan kebutuhan *data mining* [6]. Dalam penelitian ini, atribut yang dibutuhkan hanya 2 atribut yaitu Bill dan Nama_Menu. Kedua atribut ini digunakan untuk memenuhi tujuan awal dimana akan dicari pola pembelian konsumen berdasarkan menu yang dibeli. Bill digunakan untuk membedakan satu transaksi dengan transaksi lainnya, dan atribut Nama_Menu digunakan untuk mengetahui menu apa saja yang dibeli dalam satu transaksi. Berikut pada tabel 3.2 merupakan atribut yang dipilih:

Tabel 1.2 Pemilihan Atribut

Bill	Nama_Menu
181200001	Espresso
181200002	Mie Goreng Gila
181200002	Milkshake Strawberry
181200003	Cappuccino
181200003	Pisang Keju
181200004	Geprek Hejo
181200004	Geprek Mozzarella
181200004	Ice Tea
181200004	Thai Lychee Tea
181200005	Nasi Goreng Spesial
181200005	Katsudon
181200005	Ice Tea
181200006	Ayam Bakar
181200006	Ice Tea
181200007	Hot Crispy Chicken
181200007	Ayam Bakar
181200007	Ice Tea
181200008	Flat Wine
.....	
101200057	Nasi Goreng Spesial

2. *Clean Data*

Setelah pemilihan data, selanjutnya adalah proses pembersihan data. Pembersihan data ini merupakan proses menghilangkan *noise* pada data. Dalam penelitian ini *noise* berarti data transaksi yang memiliki data tunggal. Berikut adalah tabel sebelum pembersihan dan sesudah pembersihan dapat dilihat pada tabel 3.3 dan 3.4 :

Tabel 1.3 Transaksi sebelum pembersihan

Bill	Nama_Menu
181200001	Espresso
181200002	Mie Goreng Gila
181200002	Milkshake Strawberry
181200003	Cappuccino
181200003	Pisang Keju
181200004	Geprek Hejo
181200004	Geprek Mozzarella
181200004	Ice Tea
181200004	Thai Lychee Tea
181200005	Nasi Goreng Spesial
181200005	Katsudon
181200005	Ice Tea
181200006	Ayam Bakar
181200006	Ice Tea
181200007	Hot Crispy Chicken
181200007	Ayam Bakar
181200007	Ice Tea
181200008	Flat Wine

Tabel 1.4 Transaksi setelah pembersihan

Bill	Nama_Menu
181200002	Mie Goreng Gila
181200002	Milkshake Strawberry
181200003	Cappuccino
181200003	Pisang Keju
181200004	Geprek Hejo
181200004	Geprek Mozzarella
181200004	Ice Tea
181200004	Thai Lychee Tea
181200005	Nasi Goreng Spesial
181200005	Katsudon
181200005	Ice Tea
181200006	Ayam Bakar
181200006	Ice Tea
181200007	Hot Crispy Chicken
181200007	Ayam Bakar
181200007	Ice Tea

3.1.2.3. *Modeling*

Pemodelan atau desain sistem yang akan dibangun pada tahap ini bertujuan untuk menemukan pola asosiasi pembelian konsumen, maka dari itu teknik yang digunakan adalah *Association Rule* dengan menggunakan Algoritma CT-Pro. *Association Rule* terbagi menjadi dua tahap pengerjaan, yaitu mencari frekuensi tinggi item dan *generate rule* [16]. Jika menggunakan Algoritma CT-Pro, tahap pengerjaan *Association rules* diuraikan sebagai berikut [16] :

1. Mencari *frequent itemset*
 - a. Menghitung jumlah kemunculan tiap menu
 - b. Seleksi dengan menentukan nilai minimum *support*
 - c. Membangun *Global Item Table*
 - d. *Mapping Data*
 - e. Pembentukan *Global CFP-Tree*
 - f. Pembentukan *Local CFP-Tree*
2. *Generate Rule*.

1. Mencari *Frequent Itemset*

- a. Menghitung jumlah kemunculan tiap menu

Menghitung kemunculan setiap menu merupakan tahapan pertama dalam tahapan metodologi dasar *Association Rule*. Hasil dari perhitungan kemunculan setiap menu dapat dilihat pada tabel 3.5 dibawah :

Tabel 1.5 Hasil Perhitungan tiap menu

Nama_menu	<i>Support Count</i>
Ice Tea	24
Ayam Bakar	12
Espresso	1
Thai Lychee Tea	8
Nasi Goreng Spesial	9
Geprek Mozzarella	1

Nama_menu	Support Count
Katsudon	9
GepreK Merah	1
Hot Crispy Chicken	8
Milky Caramelly	1
Latte	5
Milkshake Strawberry	1
Cappuccino	1
Geprek Hejo	5
Ice Cream	4
Mie Goreng Gila	1
Pisang Keju	1
Milkshake Chocolate	4
Roti Bakar	1
Greentea Caramelly	1

b. Seleksi dengan menentukan nilai *minimum support*

Setelah dihitung jumlah kemunculan masing-masing setiap menu maka selanjutnya akan ditentukan nilai *minimum support* untuk melihat batasan terendah munculnya menu. Dalam menentukan nilai *minimum support* dapat dilakukan sesuai dengan kebutuhan *user*. Karena, jika nilai *minimum support* 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 ≥ 2 kali. Untuk menu yang *support count*-nya tidak memenuhi batasan *minimum support* akan dieliminasi. Berikut pada tabel 3.6 adalah hasil eliminasi kemunculan menu yang tidak memenuhi *minimum support* :

Tabel 1.6 Menu yang memenuhi minimum support

Nama_menu	Support Count
Ice Tea	24
Ayam Bakar	12
Nasi Goreng Spesial	9
Katsudon	9
Thai Lychee Tea	8
Hot Crispy Chicken	8
Geprek Hejo	5
Latte	5
Ice Cream	4
Milkshake Chocolate	4

Untuk menu yang *count*-nya tidak memenuhi batas *minimum support* akan dieliminasi dan berikut data transaksi penjualan sesudah pengeliminasian menu yang tidak memenuhi *minimum support* dapat dilihat pada tabel 3.7 di bawah ini:

Tabel 1.7 Data transaksi penjualan sesudah pengeliminasian

Bill	Nama_Menu	Bill	Nama_Menu
181200004	Geprek Hejo	181200015	Katsudon
181200004	Ice Tea	181200015	Ice Tea
181200004	Thai Lychee Tea	181200016	Ayam Bakar
181200005	Nasi Goreng Spesial	181200016	Ice Tea
181200005	Katsudon	181200016	Ice Cream
181200005	Ice Tea	181200018	Latte
181200006	Ayam Bakar	181200018	Milkshake Chocolate
181200006	Ice Tea	181200018	Ice Cream
181200007	Hot Crispy Chicken	181200020	Ayam Bakar
181200007	Ayam Bakar	181200020	Milkshake Chocolate
181200007	Ice Tea	181200020	Ice Tea
181200010	Ayam Bakar	181200021	Geprek Hejo
181200010	Ice Tea	181200021	Latte
181200012	Katsudon	181200021	Thai Lychee Tea
181200012	Ice Tea	181200021	Ice Cream
181200013	Nasi Goreng Spesial	181200023	Katsudon
181200013	Milkshake Chocolate	181200023	Ice Tea

Bill	Nama_Menu
181200024	Nasi Goreng Spesial
181200024	Ice Tea
181200024	Hot Crispy Chicken
181200025	Hot Crispy Chicken
181200025	Ayam Bakar
181200027	Hot Crispy Chicken
181200027	Ayam Bakar
181200027	Ice Tea
181200027	Milkshake Chocolate
181200028	Katsudon
181200028	Thai Lychee Tea
181200029	Katsudon
181200029	Ice Tea
181200032	Ice Tea
181200032	Katsudon
181200032	Ayam Bakar
181200033	Latte
181200033	Ayam Bakar
181200035	Nasi Goreng Spesial
181200035	Ice Tea
181200036	Ayam Bakar
181200036	Thai Lychee Tea
181200036	Ice Tea
181200038	Ice Tea
181200038	Nasi Goreng Spesial
181200039	Ice Tea
181200039	Nasi Goreng Spesial

Bill	Nama_Menu
181200040	Hot Crispy Chicken
181200040	Latte
181200040	Thai Lychee Tea
181200041	Katsudon
181200041	Ice Cream
181200041	Ice Tea
181200043	Geprek Hejo
181200043	Thai Lychee Tea
181200044	Geprek Hejo
181200044	Thai Lychee Tea
181200044	Hot Crispy Chicken
181200045	Katsudon
181200045	Thai Lychee Tea
181200046	Ayam Bakar
181200046	Hot Crispy Chicken
181200047	Nasi Goreng Spesial
181200047	Ice Tea
181200049	Latte
181200049	Ice Tea
181200052	Ayam Bakar
181200052	Ice Tea
181200053	Geprek Hejo
181200053	Nasi Goreng Spesial
181200053	Ice Tea
181200055	Nasi Goreng Spesial
181200055	Ice Tea
181200055	Hot Crispy Chicken

c. Membangun *Global Item Table*

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

Tabel 1.8 Global item table

Nama_Menu	Support Count	Id Global Item
Ice Tea	24	1
Ayam Bakar	12	2
Katsudon	9	3
Nasi Goreng Spesial	9	4
Hot Crispy Chicken	8	5
Thai Lychee Tea	8	6
Latte	5	7
Geprek Hejo	5	8
Ice Cream	4	9
Milkshake Chocolate	4	10

d. *Mapping Data*

Setelah Membangun *Global Item* tabel selanjutnya yaitu *Mapping id global item* ke data transaksi dari tabel 3.7, di mana Nama_menu pada tabel 3.7 digantikan oleh *id Global item*. Untuk hasil pemberian *id Global Item* dapat dilihat pada tabel 3.9 dibawah ini :

Tabel 1.9 Mapping Id global item ke data transaksi

Bill	Nama_Menu	Bill	Nama_Menu
181200004	8	181200012	1
181200004	1	181200013	4
181200004	6	181200013	10
181200005	4	181200015	3
181200005	3	181200015	1
181200005	1	181200016	2
181200006	2	181200016	1
181200006	1	181200016	9
181200007	5	181200018	7
181200007	2	181200018	10
181200007	1	181200018	9
181200010	2	181200020	2
181200010	1	181200020	10
181200012	3	181200020	1

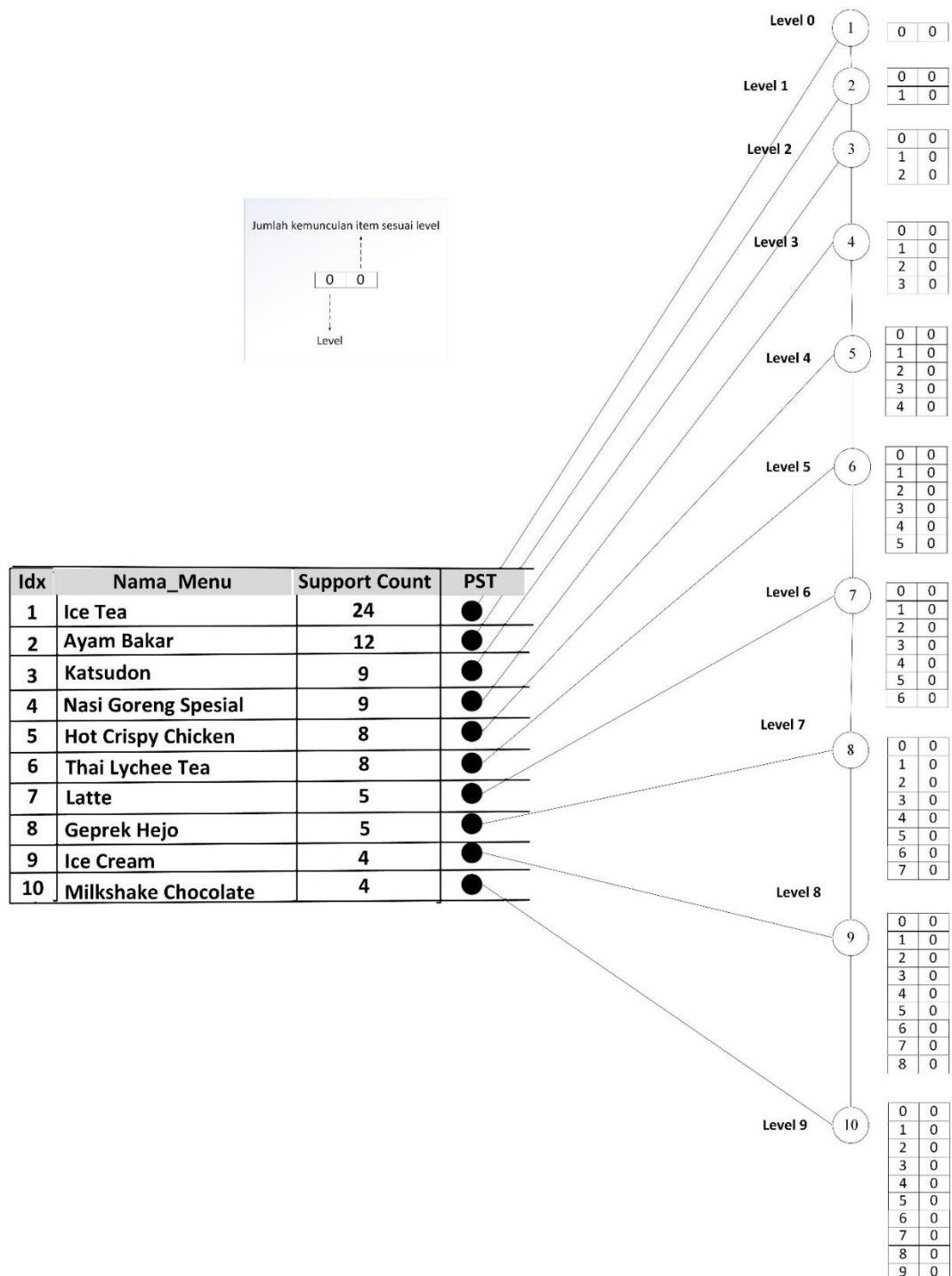
Bill	Nama_Menu
181200021	8
181200021	7
181200021	6
181200021	9
181200023	3
181200023	1
181200024	4
181200024	1
181200024	5
181200025	5
181200025	2
181200027	5
181200027	2
181200027	1
181200027	10
181200028	3
181200028	6
181200029	3
181200029	1
181200032	1
181200032	3
181200032	2
181200033	7
181200033	2
181200035	4
181200035	1
181200036	2
181200036	6
181200036	1
181200038	1
181200038	4
181200039	1
181200039	4
181200040	5
181200040	7
181200040	6
181200041	3
181200041	9
181200041	1
181200043	8

Bill	Nama_Menu
181200043	6
181200044	8
181200044	6
181200044	5
181200045	3
181200045	6
181200046	2
181200046	5
181200047	4
181200047	1
181200049	7
181200049	1
181200052	2
181200052	1
181200053	8
181200053	4
181200053	1
181200055	4
181200055	1
181200055	5

e. Pembentukan CFP-Tree

Setelah proses *Mapping data* selesai kemudian tahap selanjutnya adalah membangun *Compressed Fp-Tree* dan memasukkan setiap data transaksinya ke dalam *tree*, CFP-Tree memiliki *root* yang mewakili *index* dari *item* dengan tingkat kemunculan tertinggi. Berikut ini adalah beberapa hal yang harus diperhatikan pada saat membangun CFP-Tree:

1. Tahapan pertama dalam pembentukan CFP-Tree adalah dengan membuat *left branch* yaitu dengan men-*generate node* mulai dari *index* ke-1 sampai ke-n. Berikut dibawah ini pada gambar 3.2 merupakan *leftmost branch* yang terbentuk :

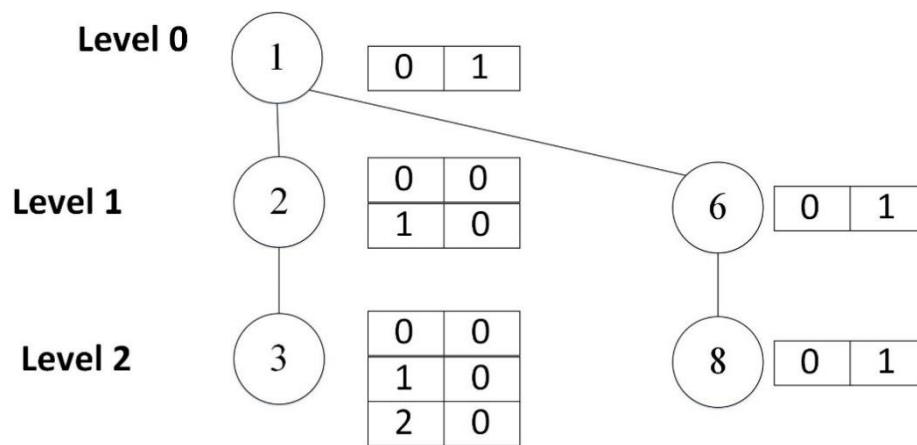


Gambar 1.2 leftmost branch yang terbentuk

- Masukkan transaksi kedalam *tree* dimulai dari *item* yang muncul dari nomor transaksi pertama yang telah di *mapping*. Hitung jumlah kemunculan pada *item* yang muncul sesuai dengan levelnya atau dengan *id parent*-nya.

3. Jika *node* sudah memiliki anak, kemudian *item* yang muncul selanjutnya sama dengan anak tersebut maka tambahkan jumlah kemunculan =1 pada level atau *parent* yang sama.
4. Jika *node* sudah memiliki anak, tetapi *item* yang muncul selanjutnya berbeda dengan anak dari *node* tersebut, maka buat *node* baru dan tambah jumlah kemunculan = 1 pada level atau *parent* yang sama, hubungkan *node* baru tersebut pada induknya dengan garis (*nodelink*).
- i. CFP-Tree bill 181200004

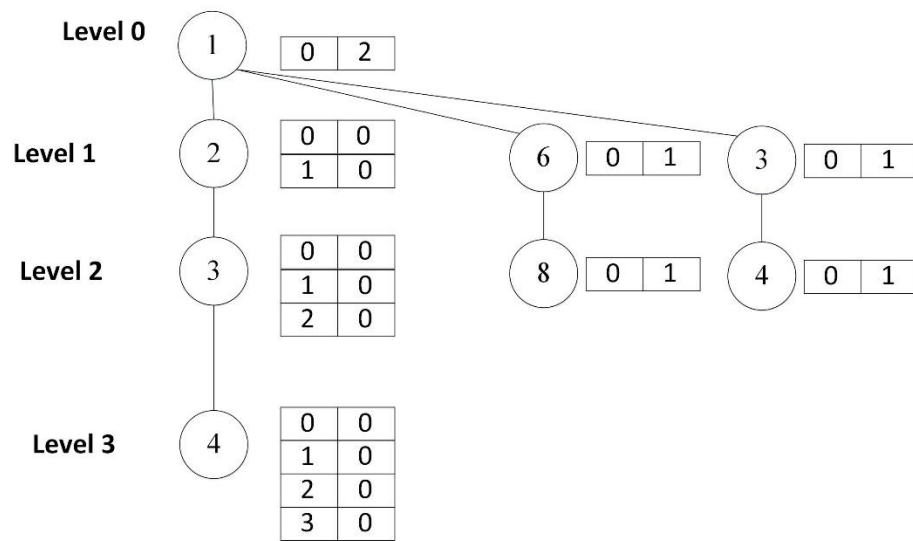
Pada bill 181200004 terdapat *index* 1, 6, 8. *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, buat *node* baru dengan *index* 6 dan *index* 8 sebagai anak dari *node index* 1, tambah 1 pada jumlah kemunculan atau *count*-nya atau dapat dilihat pada gambar 3.3 berikut:



Gambar 1.3 CFP-Tree bill 181200004

- ii. CFP-Tree bill 181200005

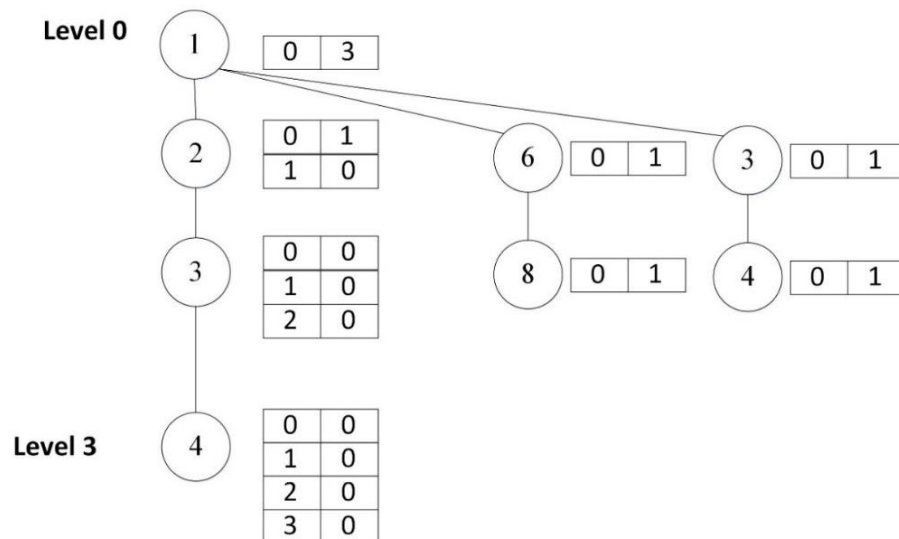
Pada bill 181200005 terdapat *index* 1, 3, 4. *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, buat *node* baru dengan *index* 3 dan *index* 4 sebagai anak dari *node index* 1. Tambah jumlah kemunculan atau *count* 1 atau dapat dilihat pada gambar 3.4 berikut:



Gambar 1.4 CFP-Tree bill 181200005

iii. CFP-Tree bill 181200006

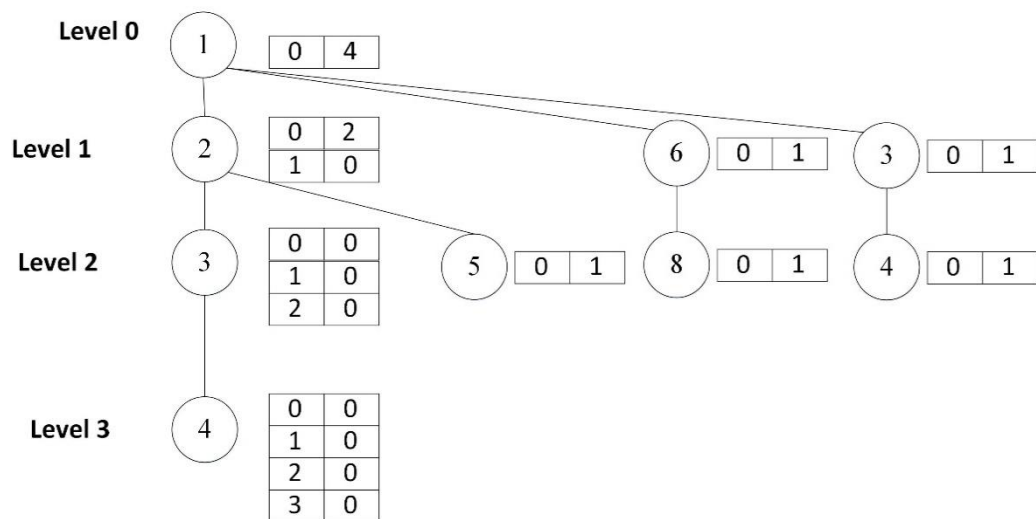
Pada bill 181200006 terdapat *index* 1, 2. *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, jumlah kemunculan *index* 2 pada level 0 atau *parent-id* 0 juga bertambah 1 atau dapat dilihat pada gambar 3.5 berikut:



Gambar 1.5 CFP-Tree bill 181200006

iv. CFP-Tree bill 181200007

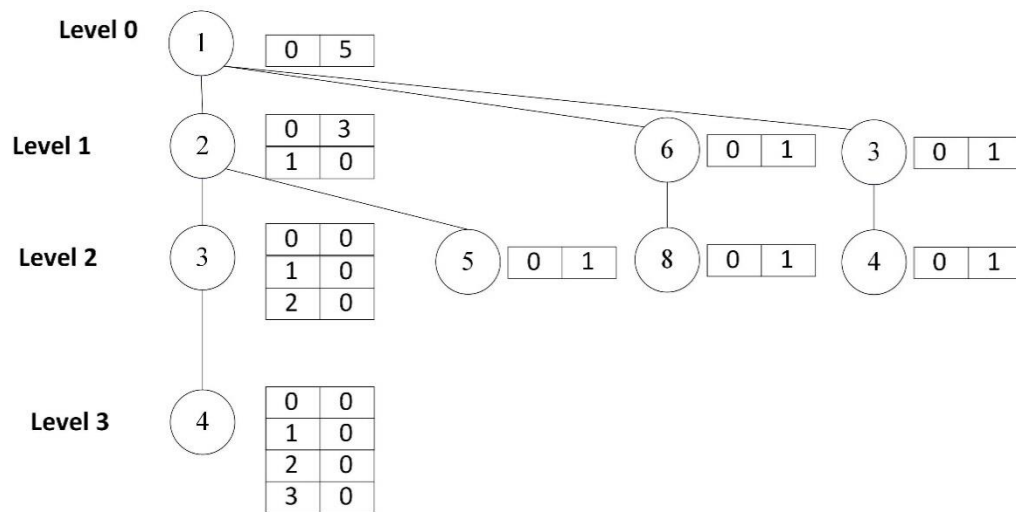
Pada bill 181200007 terdapat *index* 1, 2, 5. *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, jumlah kemunculan *index* 2 pada level 0 atau *parent-id* 0 juga akan bertambah 1. Buat *node* baru *index* 5 yang terhubung dengan *index* 2 sebagai anak dari *index* 1 dan tambah jumlah kemunculan 1 atau dapat dilihat pada gambar 3.6 berikut:



Gambar 1.6 CFP-Tree bill 181200007

v. CFP-Tree bill 181200010

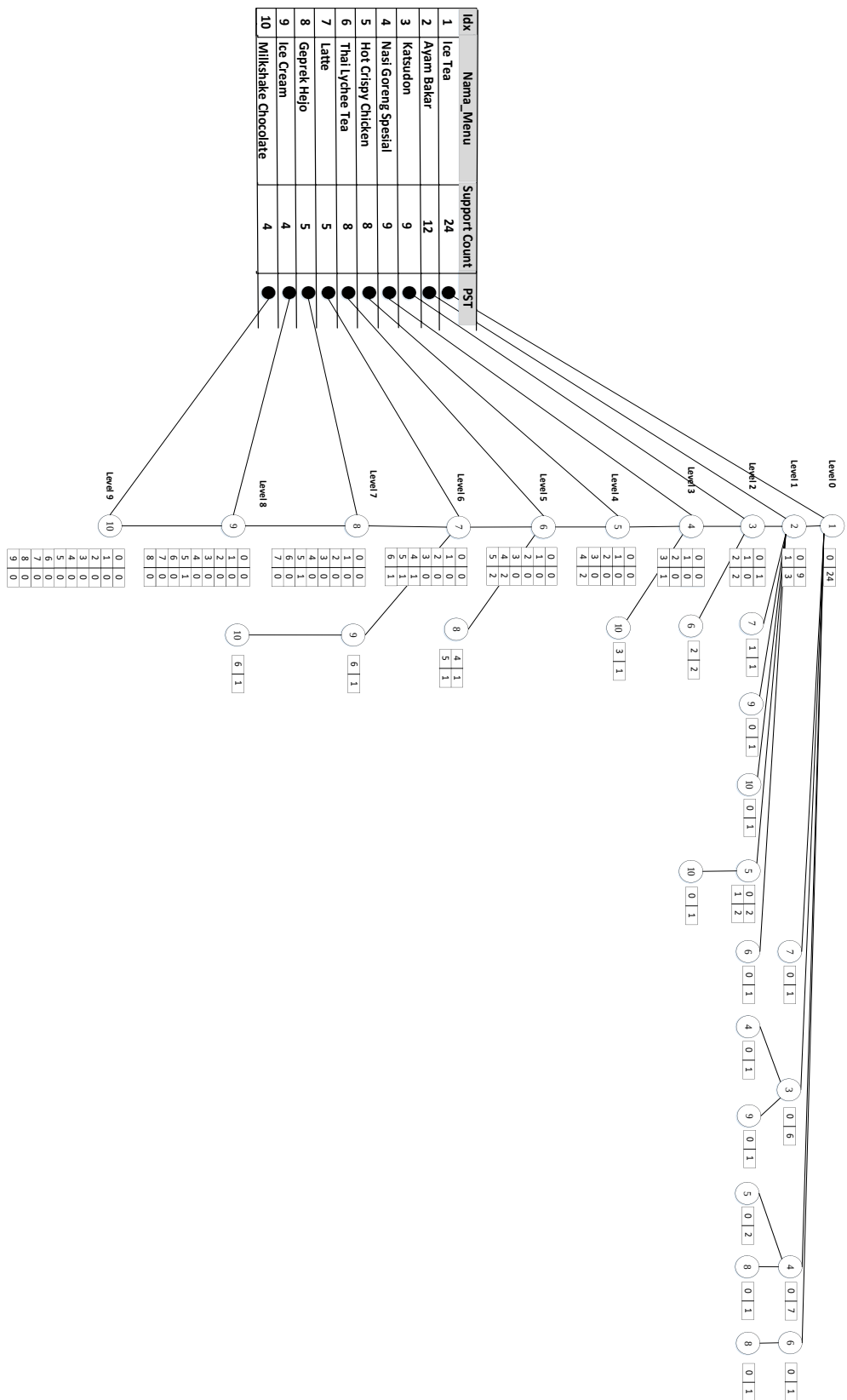
Pada bill 181200010 terdapat *index* 1, 2. *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, jumlah kemunculan *index* 2 pada level 0 juga akan bertambah 1 atau dapat dilihat pada gambar 3.7 berikut:



Gambar 1.7 CFP-Tree bill 181200010

vi. CFP-Tree bill 181200055

Pada bill terakhir yaitu bill 181200055 terdapat *index* 1, 4, 5 . *Index* yang pertama muncul adalah 1 dimana *index* 1 berada pada level 0. Jumlah kemunculan pada *index* 1 akan bertambah 1, jumlah kemunculan *index* 4 serta *index* 5 pada level 0 yang merupakan anak *index* 1 juga akan bertambah 1 atau dapat dilihat pada gambar 3.8 berikut:



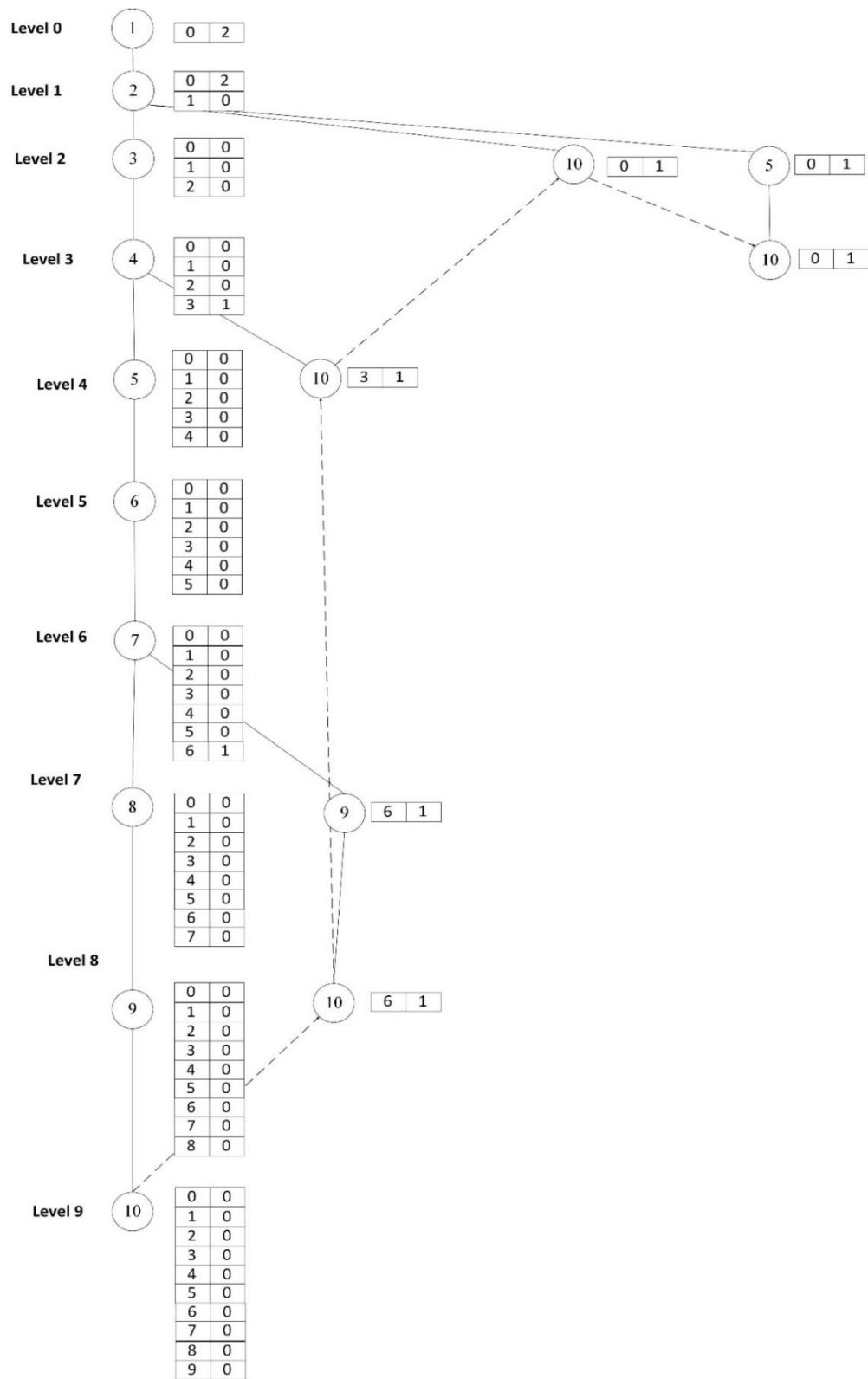
Gambar 1.8 CFP-Tree bill terakhir

f. Pembentukan *Local CFP-Tree*

Setelah tahap pembangunan CFP-Tree dari data transaksi, selanjutnya adalah membangun *local CFP-Tree* dari setiap *index* untuk mencari *frequent itemset* yang lebih akurat. Langkah pembuatan *Local CFP-Tree* sama dengan pembuatan *Global CFP-Tree* yang membedakan hanya jika pembuatan *Global CFP-Tree* dibangun dari *Global Item Table* sedangkan *Local CFP-Tree* dibangun dari tabel *local Frequent Pattern* dari setiap *index*.

1) *Local CFP-Tree index 10*

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 10*, yang mana pencarian *local frequent pattern index 10* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhiran di *index 10* dari *node* paling akhir sampai ke *root*. Berikut ini pada gambar 3.9 adalah hasil penelusuran *Global CFP-Tree* yang berakhiran *index 10*:



Gambar 1.9 Global CFP-Tree index 10

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 10* :

Tabel 1.10 *local frequent pattern index 10*

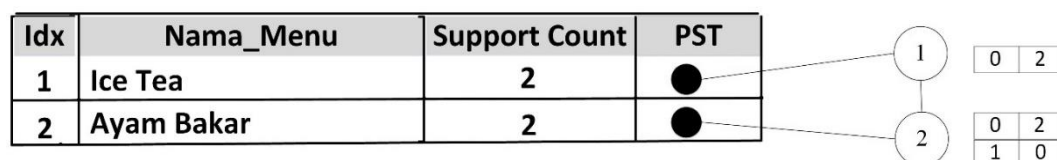
<i>Index</i>	<i>Local Frequent Pattern</i>	<i>Node memenuhi minimum support</i>
10	1,2 : 1 1,2,3 : 1 4 : 1 7, 9 : 1	1 : 2 2 : 2

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap *Nama_menu* berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.11 *local item table index 10*

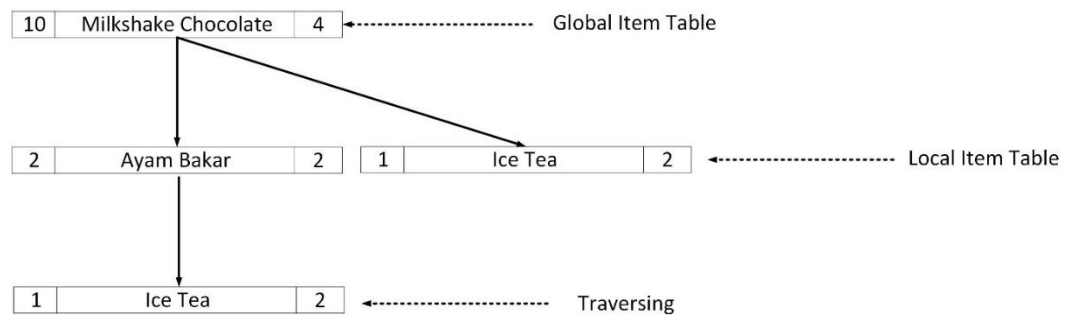
<i>Idx local</i>	<i>Idx global</i>	<i>Nama_menu</i>	<i>Count</i>
1	1	Ice Tea	2
2	2	Ayam Bakar	2

Setelah *Local item table* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini pada gambar 3.10 adalah *Local CFP-Tree* yang terbentuk :



Gambar 1.10 *Local CFP-Tree index 10*

Setelah *Local CFP-Tree index 10* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* dimana *index 10* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 10* atau dapat dilihat pada gambar 3.11 berikut ini :

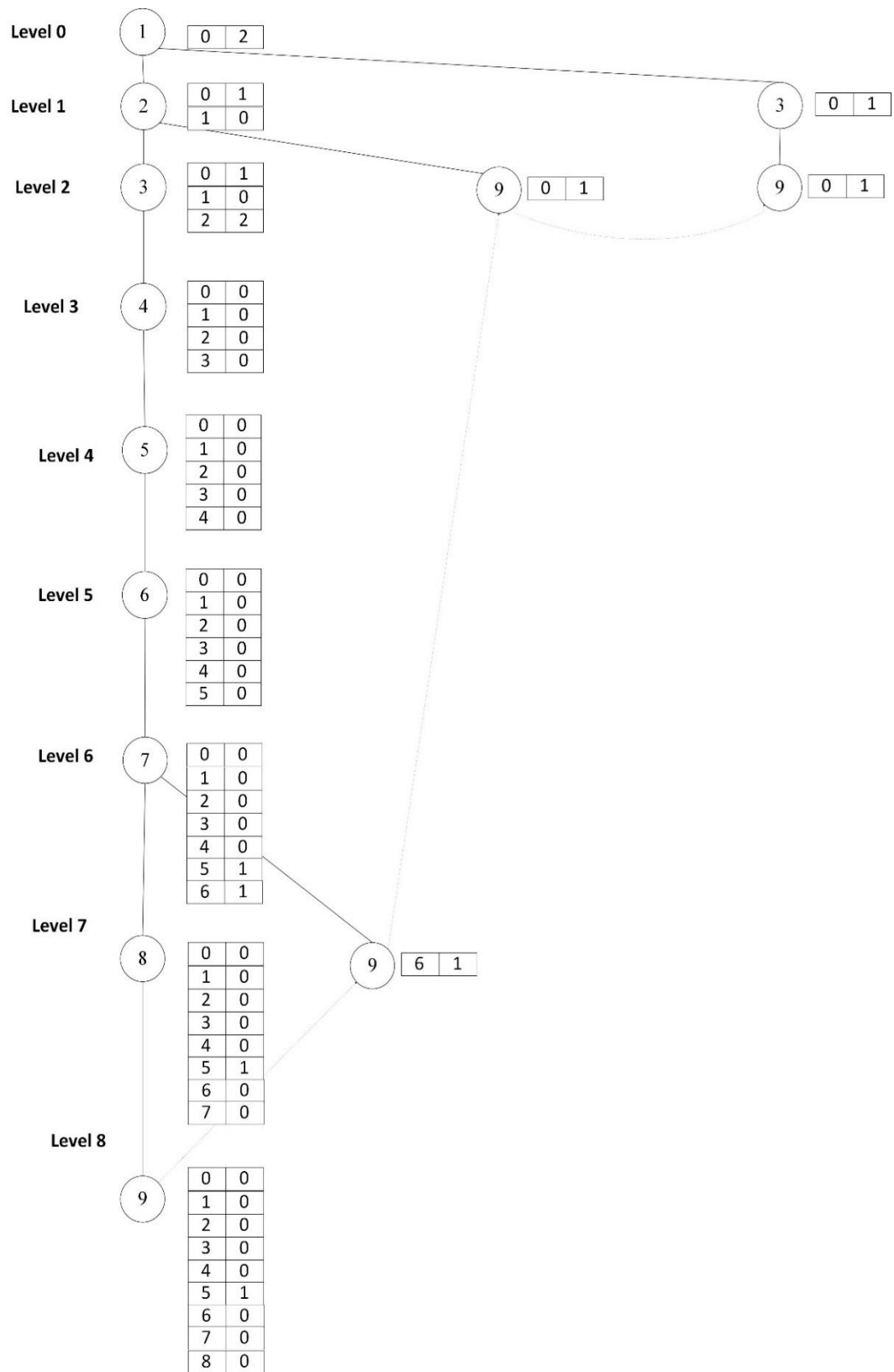


Gambar 1.11 Local CFP-Tree projection index 10

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index 10* dengan Nama_Menu Milkshake Chocolate adalah sebagai berikut: (10,2:2), (10,2,1:2), (10,1:2).

2) Local CFP-Tree index 9

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 9*, yang mana pencarian *local frequent pattern index 9* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhiran di *index 9* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhiran *index 9*:



Gambar 1.12 Global CFP-Tree index 9

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 9*:

Tabel 1.12 *local frequent pattern index 9*

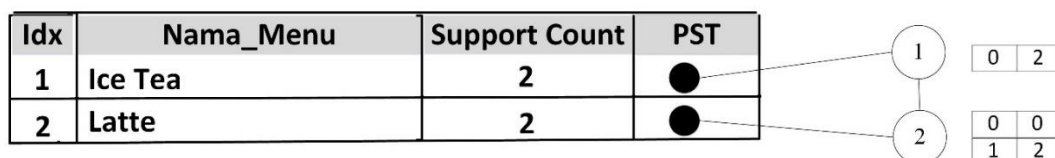
<i>Index</i>	<i>Local Frequent Pattern</i>	<i>Node memenuhi minimum support</i>
9	1,2 : 1 1,3 : 1 6,7,8 : 1 7 : 1	1 : 2 7 : 2

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap *Nama_menu* berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.13 *local item table index 9*

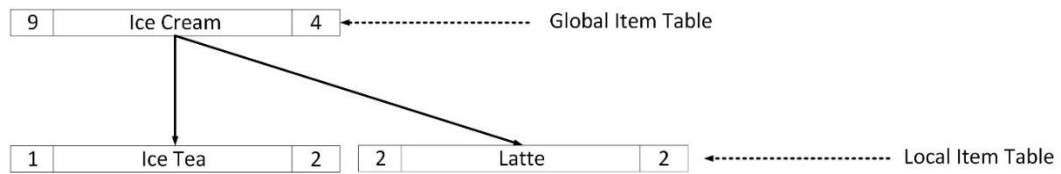
<i>Idx local</i>	<i>Idx global</i>	<i>Nama_menu</i>	<i>Count</i>
1	1	Ice Tea	2
2	7	Latte	2

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :



Gambar 1.13 *Local CFP-Tree index 9*

Setelah *Local CFP-Tree index 9* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 9* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 9* atau dapat dilihat pada gambar 3.14 berikut ini :

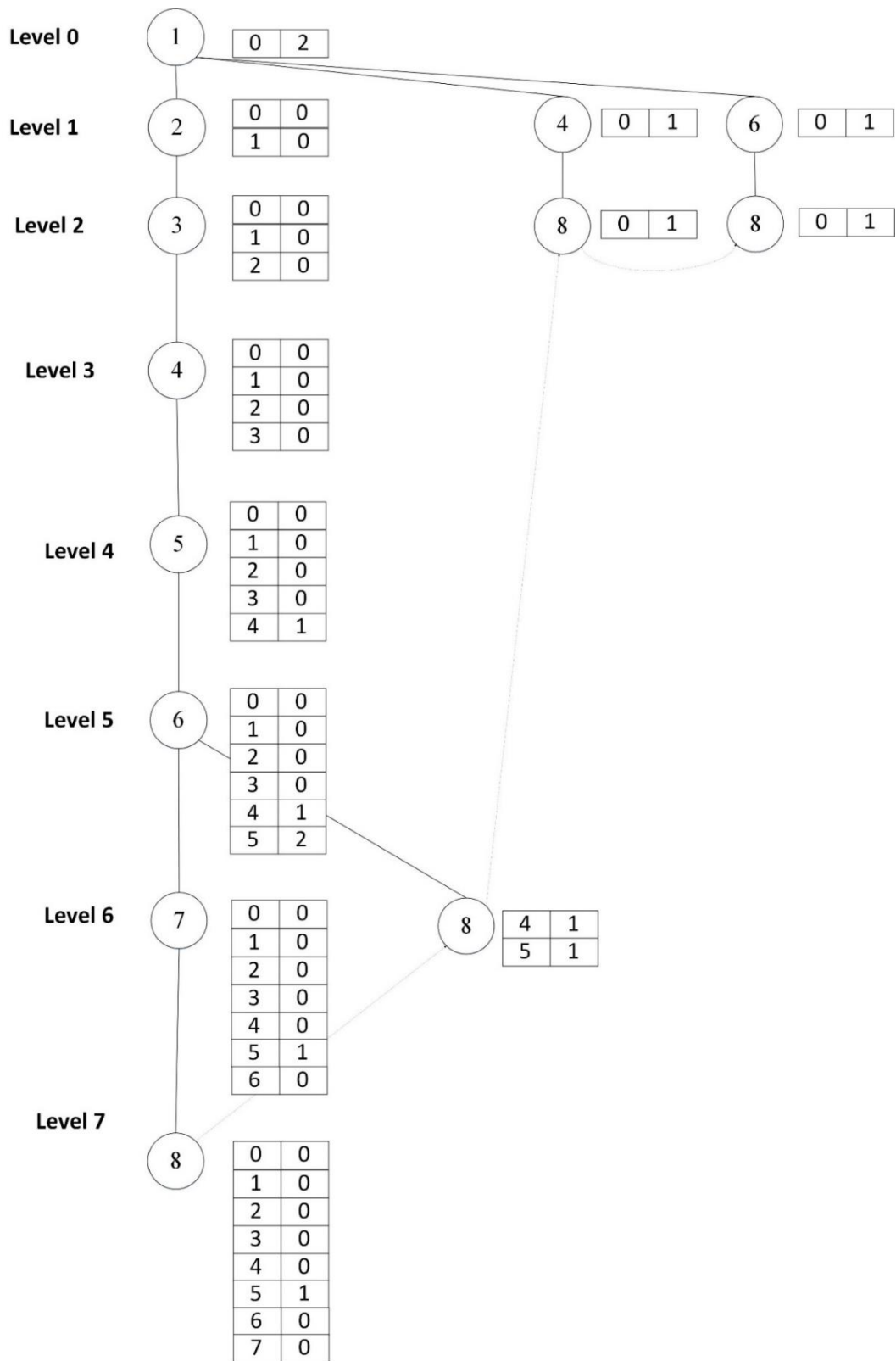


Gambar 1.14 Local CFP-Tree projection index 9

Dari *Local CFP-Tree* projection maka didapatkan *item frequent* untuk *Index* 9 dengan Nama_Menu Ice Cream adalah sebagai berikut: (9,2:2), (9,1:2).

3) *Local CFP-Tree index 8*

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index* 8, yang mana pencarian *local frequent pattern index* 8 dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhiran di *index* 8 dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhiran *index* 8:



Gambar 1.15 Global CFP-Tree index 8

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 8* :

Tabel 1.14 *local frequent pattern index 8*

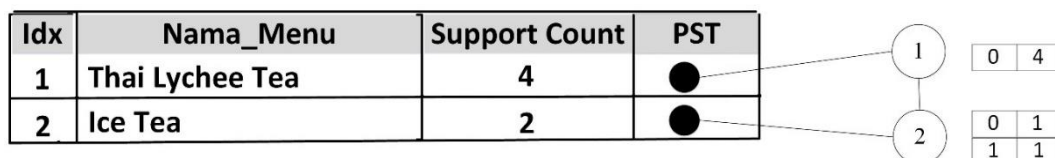
<i>Index</i>	<i>Local Frequent Pattern</i>	<i>Node memenuhi minimum support</i>
8	1,4 : 1 1,6 : 1 5,6 : 1 6 : 1 6, 7 : 1	1 : 2 6 : 4

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.15 *local item table index 8*

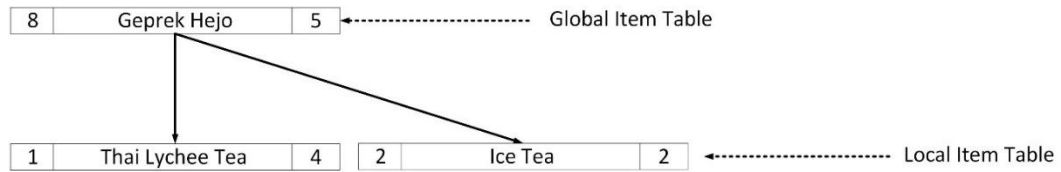
<i>Idx local</i>	<i>Idx global</i>	Nama_menu	<i>count</i>
1	6	Thai Lychee Tea	4
2	1	Ice Tea	2

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :



Gambar 1.16 *Local CFP-Tree index 8*

Setelah *Local CFP-Tree index 8* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 8* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 8*.

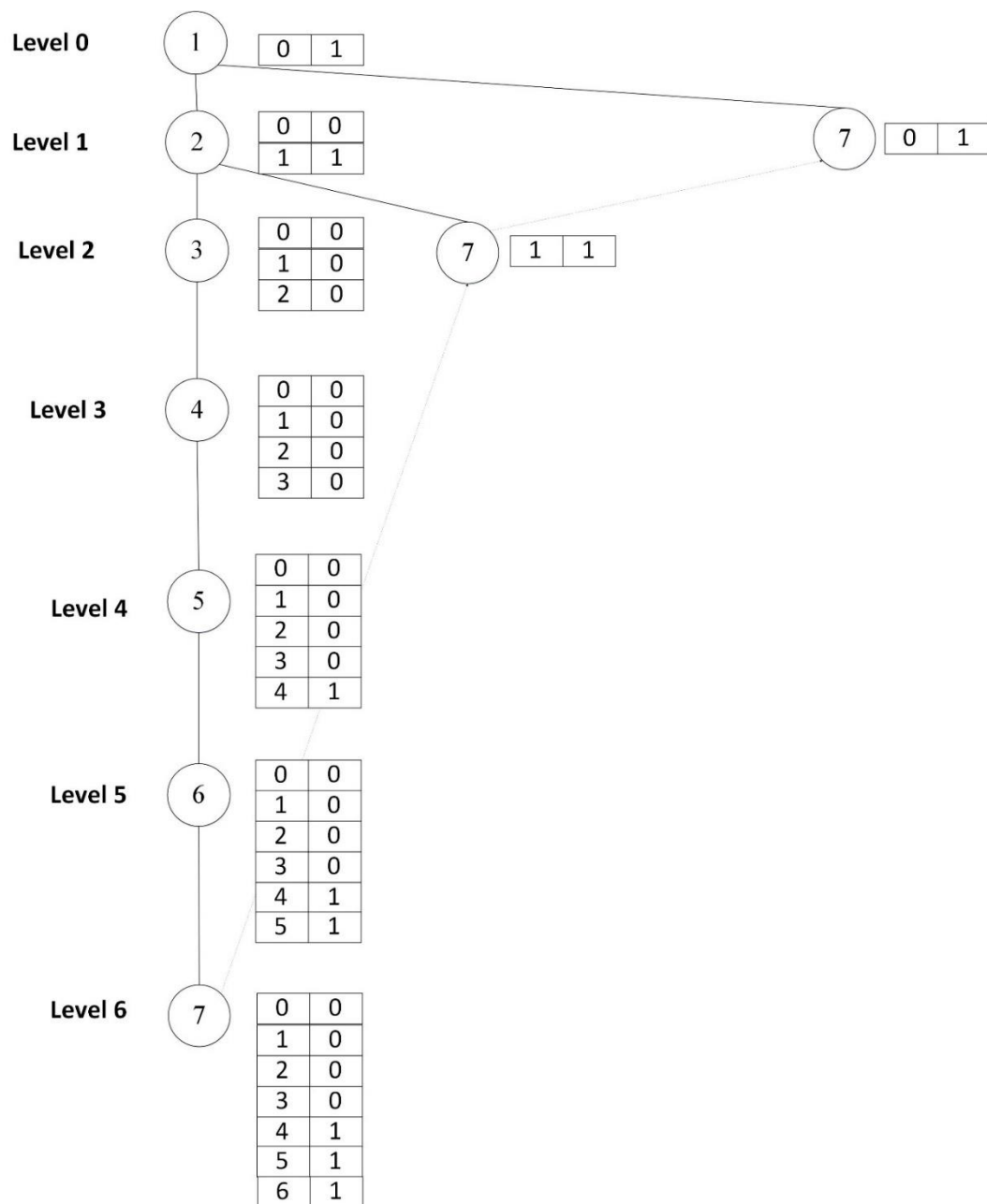


Gambar 1.17 Local CFP-Tree projection index 8

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index 8* dengan Nama_Menu Geprek Hejo adalah sebagai berikut: (8,2:2), (8,1:4).

4) *Local CFP-Tree index 7*

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 7*, yang mana pencarian *local frequent pattern index 7* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhir di *index 7* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhir *index 7*:



Gambar 1.18 Global CFP-Tree index 7

Dari gambar diatas dapat dilihat *local frequent pattern* untuk index 7 :

Tabel 1.16 local frequent pattern index 7

Index	Local Frequent Pattern	Node memenuhi minimum support
7	1 : 1 2 : 1 5,6 : 1	6 : 2

Index	Local Frequent Pattern	Node memenuhi minimum support
	6 : 1	

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.17 local item table index 7

Idx local	Idx global	Nama_menu	count
1	6	Thai Lychee Tea	2

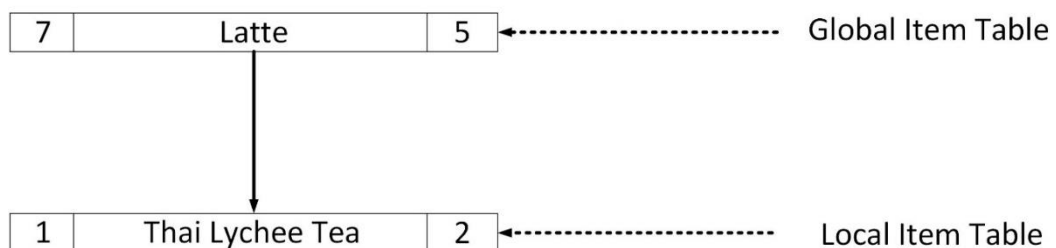
Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP -Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :

Idx	Nama_Menu	Support Count	PST
1	Thai Lychee Tea	2	●



Gambar 1.19 Local CFP-Tree index 7

Setelah *Local CFP-Tree index 7* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 7* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 7*.

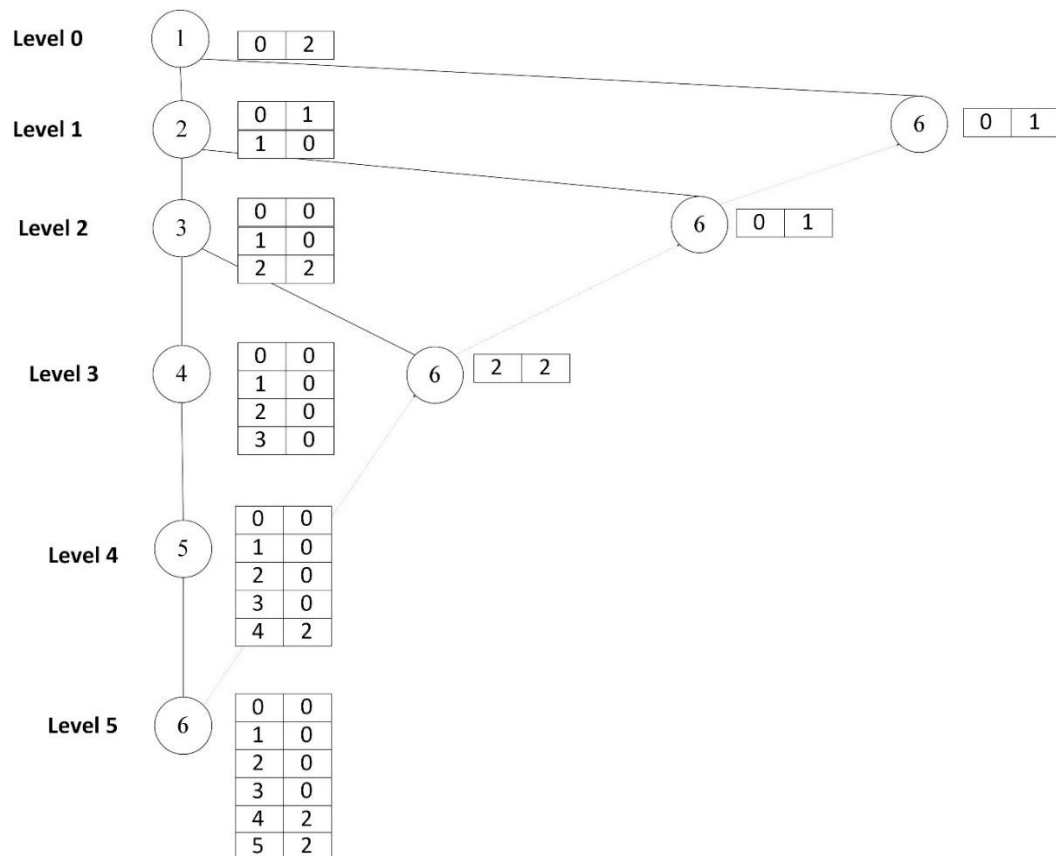


Gambar 1.20 Local CFP-Tree projection index 7

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index* 7 dengan Nama_Menu Latte adalah sebagai berikut: (7,1:2).

5) *Local CFP-Tree index 6*

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 6*, yang mana pencarian *local frequent pattern index 6* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhir di *index 6* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhir *index 6*:



Gambar 1.21 Global CFP-Tree index 6

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 6* :

Tabel 1.18 local frequent pattern index 6

Index	Local Frequent Pattern	Node memenuhi minimum support
6	1 : 1	1 : 2

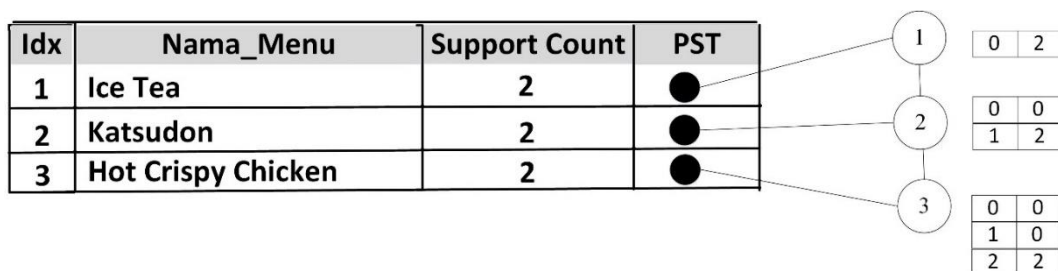
Index	Local Frequent Pattern	Node memenuhi minimum support
	1,2 : 1 3 : 2 5 : 2	3 : 2 5 : 2

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.19 local item table index 6

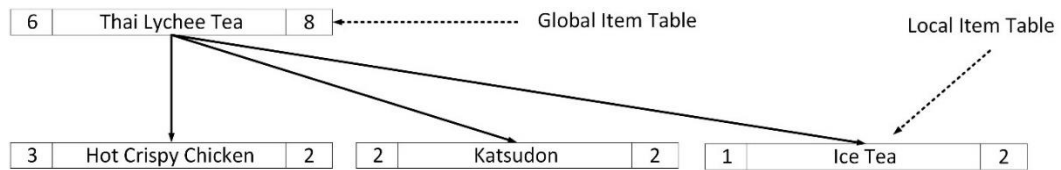
Idx local	Idx global	Nama_menu	count
1	1	Ice Tea	2
2	3	Katsudon	2
3	5	Hot Crispy Chicken	2

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP -Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :



Gambar 1.22 Local CFP-Tree index 6

Setelah *Local CFP-Tree index 6* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 6* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 6*.

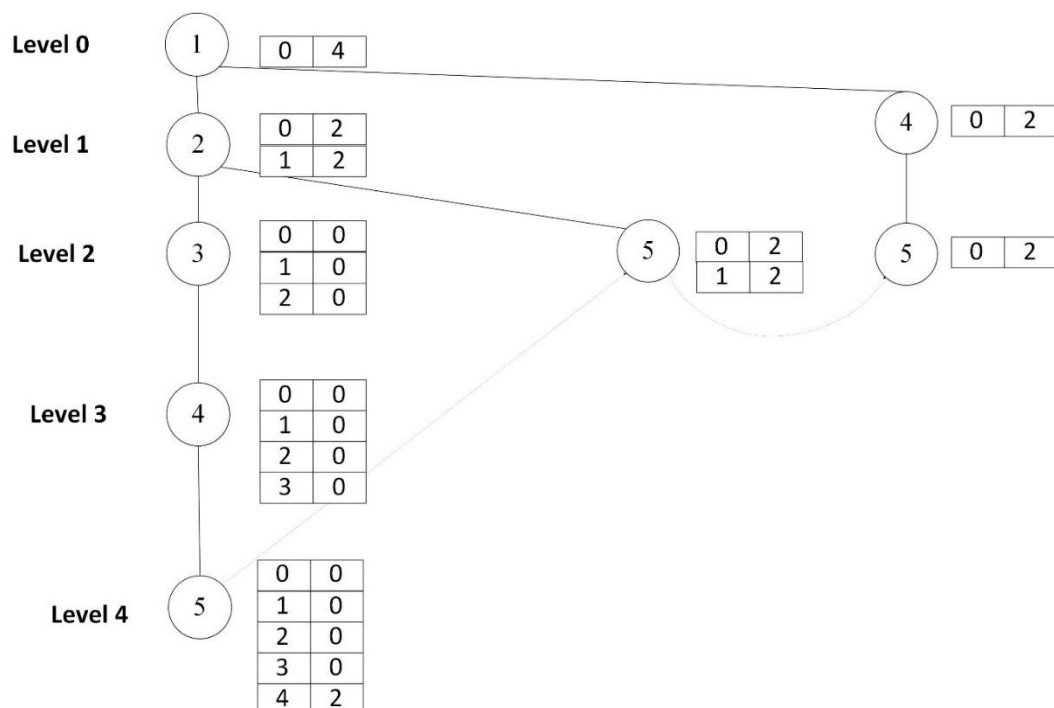


Gambar 1.23 Local CFP-Tree projection index 6

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index 6* dengan Nama_Menu Thai Lychee Tea adalah sebagai berikut: (6,3:2), (6,2:2), (6,1:2).

6) Local CFP-Tree index 5

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 5*, yang mana pencarian *local frequent pattern index 5* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhir di *index 5* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhir *index 5*:



Gambar 1.24 Global CFP-Tree index 5

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 5* :

Tabel 1.20 local frequent pattern index 5

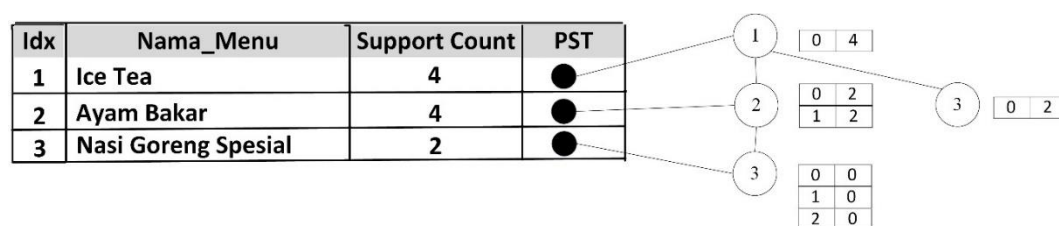
Index	Local Frequent Pattern	Node memenuhi minimum support
5	1,4 : 2	1 : 4
	1,2 : 2	2 : 4
	2 : 2	4 : 2

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

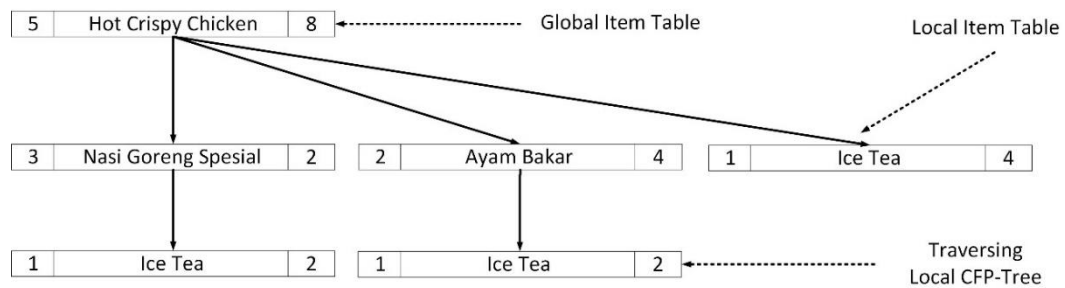
Tabel 1.21 local item table index 5

Idx local	Idx global	Nama_menu	count
1	1	Ice Tea	4
2	2	Ayam Bakar	4
3	4	Nasi Goreng Spesial	2

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :

**Gambar 1.25 Local CFP-Tree index 5**

Setelah *Local CFP-Tree index 5* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 5* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 5*.

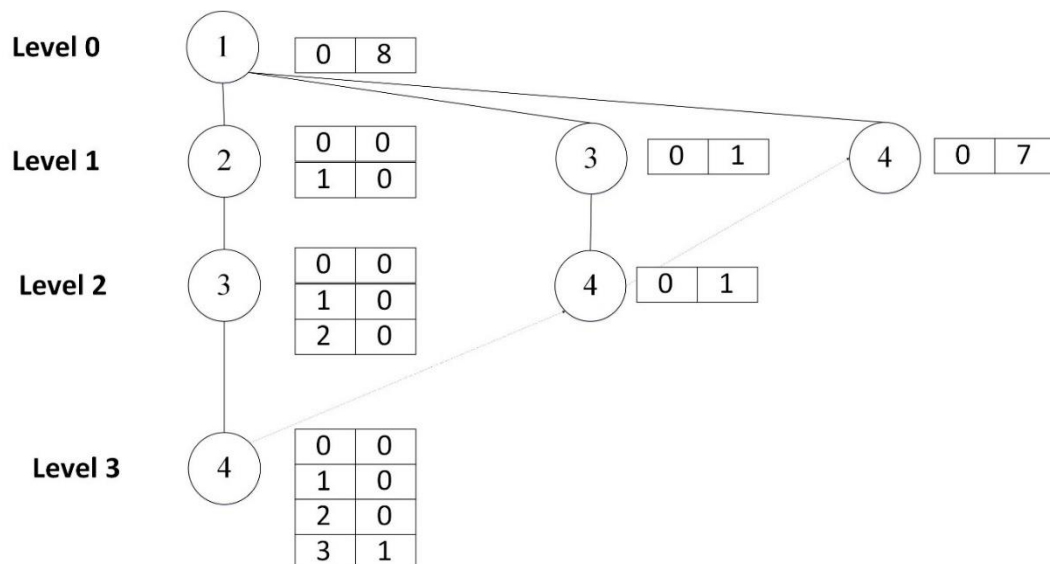


Gambar 1.26 Local CFP-Tree projection index 5

Dari *Local CFP-Tree projection* maka didapatkan *item frequent* untuk *Index 5* dengan Nama_Menu Hot Crispy Chicken adalah sebagai berikut: (5,3:2), (5,3,1:2), (5,2:4), (5,2,1:2), (5,1:4).

7) Local CFP-Tree index 4

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 4*, yang mana pencarian *local frequent pattern index 4* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhiran di *index 4* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhiran *index 4*:



Gambar 1.27 Global CFP-Tree index 4

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 4* :

Tabel 1.22 local frequent pattern index 4

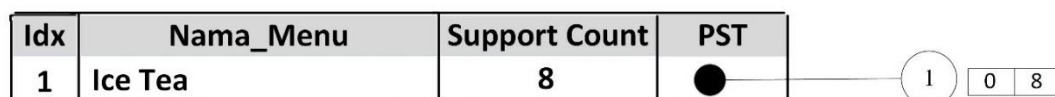
Index	Local Frequent Pattern	Node memenuhi minimum support
4	1 : 7 1,3 : 1	1 : 8

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

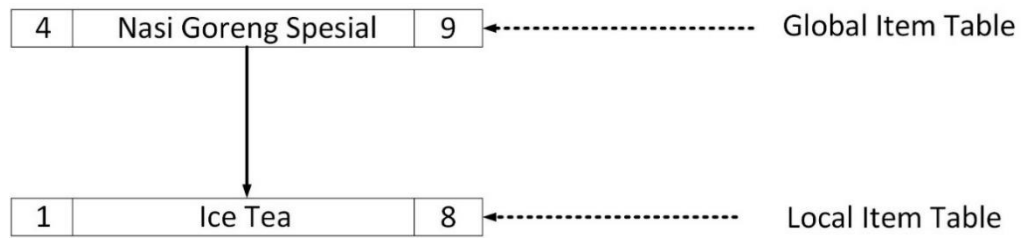
Tabel 1.23 local item table index 4

Idx local	Idx global	Nama_menu	count
1	1	Ice Tea	8

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :

**Gambar 1.28 Local CFP-Tree index 4**

Setelah *Local CFP-Tree index 4* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 4* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 4*.

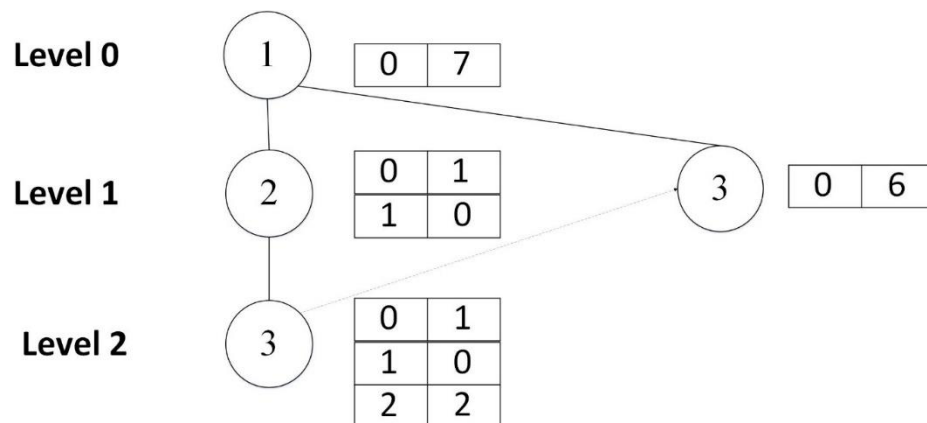


Gambar 1.29 Local CFP-Tree projection index 4

Dari *Local CFP-Tree projection* maka didapatkan item *frequent* untuk *Index 4* dengan Nama_Menu Nasi Goreng Spesial adalah sebagai berikut: (4,1:8).

8) *Local CFP-Tree index 3*

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 3*, yang mana pencarian *local frequent pattern index 3* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhiran di *index 3* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhiran *index 3*:



Gambar 1.30 Global CFP-Tree index 3

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 3* :

Tabel 1.24 local frequent pattern index 3

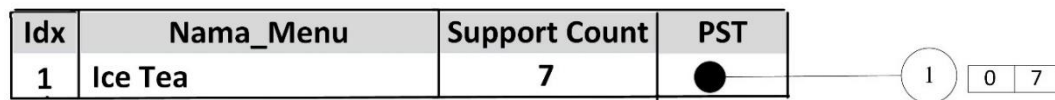
Index	Local Frequent Pattern	Node memenuhi minimum support
3	1 : 6 1,2 : 1	1 : 7

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

Tabel 1.25 *local item table index 3*

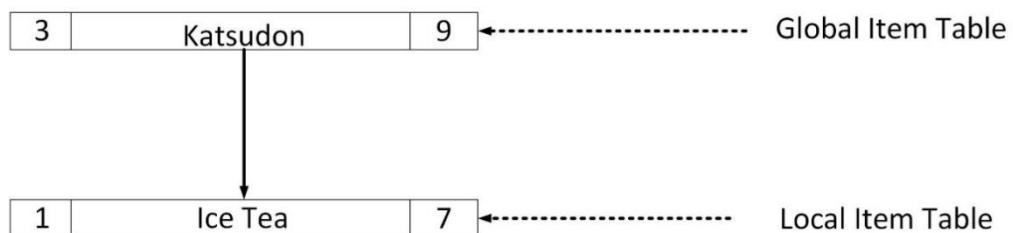
Idx <i>local</i>	Idx <i>global</i>	Nama_menu	count
1	1	Ice Tea	7

Setelah *Local item* tabel terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent pattern* untuk dimasukkan ke *local CFP-Tree*. Berikut ini adalah *Local CFP-Tree* yang terbentuk :



Gambar 1.31 *Local CFP-Tree index 3*

Setelah *Local CFP-Tree index 3* terbentuk, langkah selanjutnya adalah pembentukan *Local CFP-Tree projection* di mana *index 3* akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index 3*.

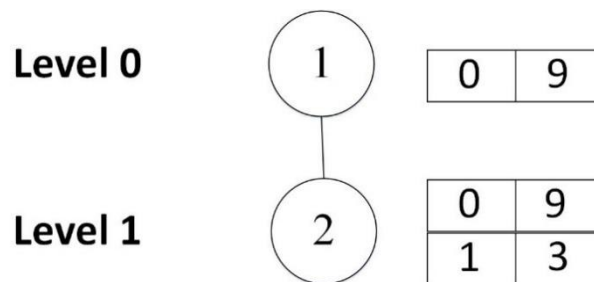


Gambar 1.32 *Local CFP-Tree projection index 3*

Dari *Local CFP-Tree projection* maka didapatkan item *frequent* untuk *Index 3* dengan Nama_Menu Katsudon adalah sebagai berikut: (3,1:7).

9) Local CFP-Tree index 2

Dari *global CFP-Tree* yang telah dibuat maka bisa dilihat *local frequent pattern* dari *index 2*, yang mana pencarian *local frequent pattern index 2* dapat dilakukan dengan menelusuri *Global CFP-Tree* yang berakhir di *index 2* dari *node* paling akhir sampai ke *root*. Berikut ini adalah hasil penelusuran *Global CFP-Tree* yang berakhir *index 2*:



Gambar 1.33 Global CFP-Tree index 2

Dari gambar diatas dapat dilihat *local frequent pattern* untuk *index 2* :

Tabel 1.26 local frequent pattern index 2

Index	Local Frequent Pattern	Node memenuhi minimum support
2	1 : 9	1 : 9

Selanjutnya, *node* yang memenuhi *minimum support* akan dimasukkan ke *local item table* dan diberikan *index* baru pada setiap Nama_menu berdasarkan jumlah kemunculan pada *local frequent pattern*. Berikut ini merupakan *local item table* yang terbentuk :

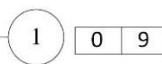
Tabel 1.27 local item table index 2

Idx local	Idx global	Nama_menu	count
1	1	Ice Tea	9

Setelah *Local item tabel* terbentuk, maka tahapan selanjutnya adalah memetakan *node* yang memenuhi nilai *minimum support* dari *local frequent*

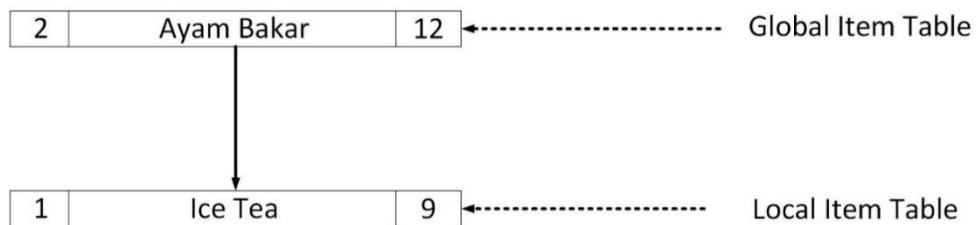
pattern untuk dimasukkan ke *local* CFP-Tree. Berikut ini adalah *Local* CFP-Tree yang terbentuk :

Idx	Nama_Menu	Support Count	PST
1	Ice Tea	9	●



Gambar 1.34 Local CFP-Tree index 2

Setelah *Local* CFP-Tree *index* 2 terbentuk, langkah selanjutnya adalah pembentukan *Local* CFP-Tree *projection* di mana *index* 2 akan menjadi *root* dan setiap *index* di *Local item* tabel akan menjadi anak dari *index* 2.



Gambar 1.35 Local CFP-Tree projection index 2

Dari *Local* CFP-Tree *projection* maka didapatkan item *frequent* untuk *Index* 2 dengan Nama_Menu Ayam Bakar adalah sebagai berikut: (2,1:9).

Setelah dilakukan pembentukan *Local* CFP-Tree pada setiap *index* maka didapatkan *frequent itemset* sebagai berikut :

Tabel 1.28 frequent itemset

Nama Menu	<i>Frequent itemset</i>
Milkshake Chocolate	• Milkshake Chocolate → Ice Tea • Milkshake Chocolate → Ayam Bakar • Milkshake Chocolate → Ayam Bakar, Ice Tea
Ice Cream	• Ice Cream → Ice Tea • Ice Cream → Latte
Geprek Hejo	• Geprek Hejo → Thai Lychee Tea • Geprek Hejo → Ice Tea
Latte	• Latte → Thai Lychee Tea

Nama Menu	<i>Frequent itemset</i>
Thai Lychee Tea	• Thai Lychee Tea → Hot Crispy Chicken • Thai Lychee Tea → Katsudon • Thai Lychee Tea → Ice Tea
Hot Crispy Chicken	• Hot Crispy Chicken → Nasi Goreng Spesial • Hot Crispy Chicken → Nasi Goreng Spesial, Ice Tea • Hot Crispy Chicken → Ayam Bakar • Hot Crispy Chicken → Ayam Bakar, Ice Tea • Hot Crispy Chicken → Ice Tea
Nasi Goreng Spesial	• Nasi Goreng Spesial → Ice Tea
Katsudon	• Katsudon → Ice Tea
Ayam Bakar	• Ayam Bakar → Ice Tea

2. Generate Rule

Dari *frequent itemset* yang didapat, akan dilakukan *generate* untuk mendapatkan *rule* yang sesuai. Dihitung dengan rumus pencarian *confidence* pada persamaan (2-4) sebagai berikut :

$$Confidence (B \rightarrow A) = \frac{\text{Jumlah transaksi mengandung } A \text{ dan } B}{\text{Jumlah transaksi mengandung } A} \quad \text{Persamaan (2-4)}$$

Hasil dari *generate* dapat dilihat pada tabel 3.29 :

Tabel 1.29 Hasil generate

<i>Frequent itemset</i>	<i>Confidence</i>
Milkshake Chocolate → Ice Tea	$2/4 * 100\% = 50\%$
Milkshake Chocolate → Ayam Bakar	$2/4 * 100\% = 50\%$
Milkshake Chocolate → Ayam Bakar, Ice Tea	$2/4 * 100\% = 50\%$
Ice Cream → Ice Tea	$2/4 * 100\% = 50\%$
Ice Cream → Latte	$2/4 * 100\% = 50\%$
Geprek Hejo → Thai Lychee Tea	$4/5 * 100\% = 80\%$
Geprek Hejo → Ice Tea	$2/5 * 100\% = 40\%$
Latte → Thai Lychee Tea	$2/5 * 100\% = 40\%$
Thai Lychee Tea → Hot Crispy Chicken	$2/8 * 100\% = 25\%$
Thai Lychee Tea → Katsudon	$2/8 * 100\% = 25\%$
Thai Lychee Tea → Ice Tea	$2/8 * 100\% = 25\%$
Hot Crispy Chicken → Nasi Goreng Spesial	$2/8 * 100\% = 25\%$
Hot Crispy Chicken → Nasi Goreng Spesial, Ice Tea	$2/8 * 100\% = 25\%$
Hot Crispy Chicken → Ayam Bakar	$4/8 * 100\% = 50\%$
Hot Crispy Chicken → Ayam Bakar, Ice Tea	$2/8 * 100\% = 25\%$
Hot Crispy Chicken → Ice Tea	$4/8 * 100\% = 50\%$
Nasi Goreng Spesial → Ice Tea	$8/9 * 100\% = 88,89\%$
Katsudon → Ice Tea	$7/9 * 100\% = 77,78\%$
Ayam Bakar → Ice Tea	$9/12 * 100\% = 75\%$

Diasumsikan *minimum confidence* 60%, karena penentuan *minimum confidence* disesuaikan semakin tinggi nilai *confidence* maka semakin kuat *rule* yang terbentuk. Berikut merupakan hasil dari *rule* dapat dilihat pada tabel 3.30 :

Tabel 1.30 Hasil Rule

<i>Frequent itemset</i>	<i>Confidence</i>
Geprek Hejo → Thai Lychee Tea	$4/5 * 100\% = 80\%$
Nasi Goreng Spesial → Ice Tea	$8/9 * 100\% = 88,89\%$
Katsudon → Ice Tea	$7/9 * 100\% = 77,78\%$
Ayam Bakar → Ice Tea	$9/12 * 100\% = 75\%$

Selanjutnya adalah pembentukan paket menu dengan harganya yang merupakan penjumlahan dari masing-masing harga menu dalam satu paket. Berikut ini pada tabel 3.31 merupakan hasil *rule* beserta harganya:

Tabel 1.31 Hasil *rule* dengan harga

Paket	Harga
Geprek Hejo → Thai Lychee Tea	Rp31.500
Nasi Goreng Spesial → Ice Tea	Rp22.000
Katsudon → Ice Tea	Rp20.000
Ayam Bakar → Ice Tea	Rp21.000

Dalam mengambil penentuan paket yang dihasilkan, kafe Kopi Cilik memiliki batasan khusus yaitu :

1. Jumlah menu yang dipaketkan 2 sampai 4 menu
2. Rentang harga paket Rp15.000 sampai Rp50.000
3. Harga paket merupakan jumlah harga kombinasi menu dikurangi Rp2.500

Dari batasan diatas, maka didapatkan rekomendasi paket pada tabel 3.32 berikut :

Tabel 1.32 pembentukan paket yang dihasilkan

Paket	Menu	Harga
Paket 1	1. Geprek Hejo 2. Thai Lychee Tea	Rp28.500
Paket 2	1. Nasi Goreng Spesial 2. Ice Tea	Rp19.500
Paket 3	1. Katsudon 2. Ice Tea	Rp17.500
Paket 4	1. Ayam Bakar	Rp18.500

Paket	Menu	Harga
	2. Ice Tea	

3.1.3. Analisis Kebutuhan Non Fungsional

3.1.3.1. Analisis Pengguna

Analisis *user* atau pengguna dibuat untuk mengetahui siapa saja dan seperti apa karakteristik yang dimiliki oleh orang yang akan menggunakan aplikasi yang akan dibuat. Adapun analisis pengguna pada penelitian ini adalah sebagai berikut:

Tabel 1.33 Tabel analisis pengguna

<i>User</i>	Tanggung jawab	Keahlian
<i>Manager</i>	Dapat mengolah data, dapat mengelola sistem yang ada, mengerti terhadap aplikasi yang akan dibangun dan dapat mengoperasikan aplikasi yang akan dibangun dengan baik.	Dapat mengoperasikan komputer dengan baik.

3.1.3.2. Analisis Perangkat Lunak

Analisis perangkat lunak dalam pembangunan aplikasi ini bertujuan untuk mengetahui spesifikasi minimum perangkat lunak yang digunakan untuk aplikasi yang akan dibangun. Berikut ini merupakan analisis terhadap perangkat lunak yang akan digunakan untuk menjalankan aplikasi yang akan dibangun di kafe Kopi Cilik.

1. Analisis perangkat lunak pada sistem yang sedang berjalan dibagian Manager kafe Kopi Cilik adalah:
 - a. Sistem Operasi Windows 10
 - b. Microsoft Office 2016
2. Analisis spesifikasi minimum perangkat lunak yang dibutuhkan untuk menjalankan sistem yang akan dibangun adalah:
 - a. Sistem Operasi Windows 7/8/10

- b. Microsoft Office 2013/2016
 - c. MySQL sebagai *database* server
3. Evaluasi kebutuhan perangkat lunak

Berdasarkan analisis spesifikasi perangkat lunak yang ada di bagian *manager* kafe Kopi Cilik, spesifikasi perangkat lunak yang digunakan masih kurang memenuhi kebutuhan. Kekurangan tersebut dapat diatasi dengan menginstal MYSQL pada komputer yang digunakan *manager*.

3.1.3.3. Analisis Perangkat Keras

Analisis kebutuhan perangkat keras dalam pembangunan aplikasi ini bertujuan untuk mengetahui spesifikasi minimum perangkat keras yang digunakan untuk aplikasi yang akan dibangun. Berikut ini merupakan analisis terhadap perangkat keras yang akan digunakan untuk menjalankan aplikasi yang akan dibangun di kafe Kopi Cilik.

1. Analisis spesifikasi perangkat keras pada sistem yang sedang berjalan di bagian Manager kafe Kopi Cilik adalah:
 - a. Processor : Intel Core i3 3.2 Ghz
 - b. Harddisk : 500 GB
 - c. Memory : 4 GB
2. Analisis spesifikasi kebutuhan perangkat keras untuk menjalankan sistem yang akan dibangun adalah:
 - a. Processor : Dual Core 1.8 GHz
 - b. Hardisk : 5 GB untuk menginstal dan menyimpan komponen yang dibutuhkan
 - c. Memory : 2 GB
3. Evaluasi kebutuhan perangkat keras

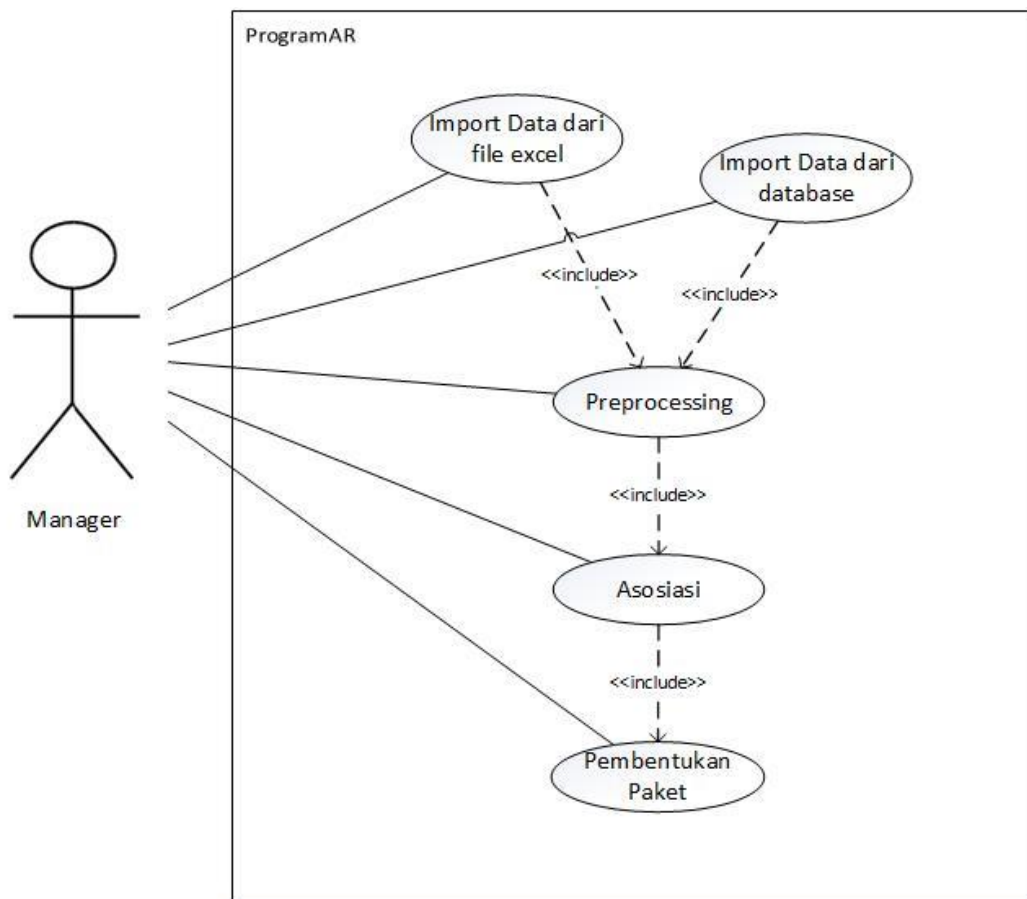
Berdasarkan analisis spesifikasi perangkat keras yang ada di bagian *Manager* kafe Kopi Cilik, perangkat keras yang digunakan sudah cukup untuk dapat menjalankan aplikasi yang akan dibangun.

3.1.4. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah analisis terhadap kebutuhan secara fungsional baik dalam aliran data ataupun informasi. Analisis kebutuhan data fungsional yang digunakan dalam sistem ini digambarkan dengan analisis pemrograman berorientasi objek dengan menggunakan konsep UML (*Unified Modeling Language*).

3.1.4.1. Use case Diagram

Diagram *use case* adalah model fungsional sebuah sistem yang menggunakan aktor dan *use case*. *Use case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk penggunaanya. Diagram *use case* dapat dilihat pada gambar 3.36 dan tabel deskripsi *use case* yang diacu pada tabel 3.34 :



Gambar 1.36 Use Case Diagram

Deskripsi *use case* :

Tabel 1.34 Deskripsi *use case*

No	<i>Use case</i>	Deskripsi
1	<i>Import</i> Data dari file excel	<i>Manager</i> harus mengimport data transaksi penjualan terlebih dahulu, data transaksi penjualan merupakan file excel. Data tersebut selanjutnya akan disimpan di <i>database</i>
2	<i>Import</i> Data dari <i>database</i>	Sistem mengambil data transaksi penjualan dari <i>database</i> kafe kopi cilik
3	<i>Preprocessing</i>	Sistem menampilkan pemilihan atribut pada data dan pembersihan data
4	Asosiasi	<i>Manager</i> menginputkan nilai minimum <i>support</i> dan minimum <i>confidence</i> untuk mencari <i>frequent itemset</i> dan <i>rule</i> yang terbentuk dari data
5	Pembentukan paket	Sistem menampilkan paket apa yang terbentuk dari hasil pembentukan <i>rule</i>

Skenario *use case* :

1. Skenario *use case Import* data dari *file* excel :

Skenario *use case import* data menggambarkan langkah aksi *user* terhadap sistem untuk melakukan *import* data transaksi penjualan yang akan disimpan ke dalam *database*. Untuk lebih jelasnya dapat dilihat pada tabel 3.35 dan 3.36 dibawah :

Tabel 1.35 Requirement A.1

<i>Requirement A.1</i>
Sistem menyediakan menu <i>import</i> data untuk melakukan <i>import</i> data transaksi penjualan ke dalam <i>database</i> sebelum melakukan asosiasi dan pembentukan paket.

Tabel 1.36 Skenario use case Import data dari file excel

<i>Use case name</i>		<i>Import Data dari file excel</i>
<i>Related Requirements</i>		A.1
<i>Goal in Context</i>		<i>Import</i> data transaksi penjualan kafe Kopi Cilik dari file excel dan menyimpannya ke dalam <i>database</i>
<i>Precondition</i>		<i>Manager</i> menyiapkan data transaksi penjualan
<i>Successful End Condition</i>		Data transaksi penjualan yang di <i>import</i> oleh <i>manager</i> berhasil disimpan ke dalam <i>database</i>
<i>Failed End Condition</i>		Data transaksi penjualan yang di <i>import</i> oleh <i>manager</i> gagal disimpan di <i>database</i>
<i>Actors</i>		<i>Manager</i>
<i>Triger</i>		<i>Manager</i> mengimport data transaksi penjualan
<i>Included Cases</i>		-
<i>Main flow</i>	<i>Step</i>	<i>Action</i>
	1	Pada menu <i>import</i> data dari file excel, <i>manager</i> menekan tombol pilih file
	2	<i>Manager</i> memilih file excel data penjualan yang akan di <i>import</i>
	3	Sistem menampilkan ekstraksi dari data yang di <i>import</i>

	4	<i>Manager</i> menekan tombol <i>import</i>
	5	Sistem menyimpan data penjualan yang dipilih ke dalam <i>database</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	2.1	Manager tidak memilih data penjualan

2. Skenario *use case Import* data dari *database* :

Skenario *use case import* data dari *database* menggambarkan langkah aksi *user* terhadap sistem untuk melakukan *import* data transaksi penjualan dari *database* kafe yang akan disimpan ke dalam *database*. Untuk lebih jelasnya dapat dilihat pada tabel 3.37 dan 3.38 dibawah :

Tabel 1.37 Requirement A.1

<i>Requirement A.1</i>
Sistem menyediakan menu <i>import</i> data untuk melakukan <i>import</i> data transaksi penjualan ke dalam <i>database</i> sebelum melakukan asosiasi dan pembentukan paket.

Tabel 1.38 Skenario *use case Import* data dari *database*

<i>Use case name</i>	<i>Import</i> Data dari <i>database</i>
<i>Related Requirements</i>	A.1
<i>Goal in Context</i>	Mengambil data transaksi penjualan kafe Kopi Cilik dari <i>database</i> kafe kopi cilik dan menyimpannya ke dalam <i>database</i>
<i>Precondition</i>	Sistem harus terhubung dengan <i>database</i> kafe
<i>Successful Condition</i> <i>End</i>	Data transaksi penjualan yang diambil oleh sistem berhasil disimpan ke dalam <i>database</i>

<i>Failed End Condition</i>		Data transaksi penjualan yang diambil oleh sistem gagal disimpan ke dalam <i>database</i>
<i>Actors</i>		<i>Manager</i>
<i>Triger</i>		<i>Manager</i> mengimport data transaksi penjualan
<i>Included Cases</i>		-
<i>Main flow</i>	<i>Step</i>	<i>Action</i>
	1	Pada menu <i>import</i> data dari <i>database</i> , <i>manager</i> menekan tombol ambil data dari <i>database</i>
	2	Sistem mengambil data dari <i>database</i> kafe
	3	Sistem menampilkan ekstraksi dari data yang di <i>import</i>
	4	<i>Manager</i> menekan tombol <i>import</i>
	5	Sistem menyimpan data penjualan yang dipilih kedalam <i>database</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>

3. Skenario *use case Preprocessing*

Skenario *use case Preprocessing* menggambarkan langkah aksi *user* terhadap sistem untuk melakukan pemilihan atribut dan pembersihan pada data. Untuk lebih jelasnya dapat dilihat pada tabel 3.39 dan 3.40 dibawah :

Tabel 1.39 Requirement A.2

<i>Requirement A.2</i>
Sistem menyediakan menu <i>Preprocessing</i> untuk melakukan pemilihan atribut dan pembersihan data.

Tabel 1.40 Skenario use case *Preprocessing*

<i>Use case name</i>		<i>Preprocessing</i>
<i>Related Requirements</i>		A.1, A.2
<i>Goal in Context</i>		Sistem melakukan pemilihan atribut pembersihan pada data
<i>Precondition</i>		Data sudah di <i>import</i> dan disimpan di <i>database</i>
<i>Successful End Condition</i>		Sistem berhasil memilih atribut dan membersihkan data
<i>Failed End Condition</i>		Sistem gagal memilih atribut dan membersihkan data
<i>Actors</i>		<i>Manager</i>
<i>Triger</i>		<i>Manager</i> melakukan <i>request</i> untuk pemilihan atribut dan pembersihan data
<i>Included Cases</i>		<i>Import data</i>
<i>Main flow</i>	<i>Step</i>	<i>Action</i>
	1	Pada menu <i>Preprocessing</i> , <i>manager</i> menekan tombol proses pemilihan atribut
	2	Sistem melakukan pemilihan atribut dan menampilkannya
	3	<i>Manager</i> menekan tombol bersihkan data
	4	Sistem membersihkan data dari <i>noise</i> dan menampilkannya
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1.1	Sistem menampilkan kesalahan gagal dalam proses pemilihan atribut

4. Skenario *use case* Asosiasi

Skenario *use case* asosiasi menggambarkan langkah aksi *user* terhadap sistem untuk melakukan asosiasi pada data transaksi penjualan yang telah di *import* dan dibersihkan oleh sistem. Untuk lebih jelasnya dapat dilihat pada tabel 3.41 dan 3.42 dibawah :

Tabel 1.41 Requirement A.3

<i>Requirement A.3</i>
Sistem menyediakan menu Asosiasi untuk melakukan pembentukan rule berdasarkan minimum <i>support</i> dan minimum <i>confidence</i> pada data transaksi penjualan yang telah di import dan telah dibersihkan dari noise.

Tabel 1.42 Skenario *use case* Asosiasi

<i>Use case name</i>	Asosiasi
<i>Related Requirements</i>	A.1, A.2, A.3
<i>Goal in Context</i>	Sistem mendapatkan <i>rule</i>
<i>Precondition</i>	Data sudah di <i>import</i> oleh <i>manager</i> dan telah melalui proses <i>preprocessing</i> oleh sistem
<i>Successful End Condition</i>	Sistem berhasil menampilkan <i>rule</i> yang terbetuk
<i>Failed End Condition</i>	Sistem gagal menampilkan <i>rule</i>
<i>Actors</i>	<i>Manager</i>
<i>Triger</i>	<i>Manager</i> memasukkan nilai minimum <i>support</i> dan minimum <i>confidence</i>
<i>Included Cases</i>	<i>Preprocessing</i>

<i>Main flow</i>	<i>Step</i>	<i>Action</i>
	1	Pada menu asosiasi, <i>manager</i> memasukkan nilai minimum <i>support</i> dan minimum <i>confidence</i>
	2	<i>Manager</i> menekan tombol proses
	3	Sistem melakukan proses pencarian <i>frequent itemset</i> sesuai nilai minimum <i>support</i>
	4	Sistem melakukan eliminasi terhadap <i>itemset</i> yang kurang dari nilai minimum <i>confidence</i>
	5	Sistem menampilkan hasil <i>rule</i>
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	2.1	<i>Minimum support</i> dan <i>minimum confidence</i> tidak diisi
	3.1	Sistem tidak mendapatkan <i>frequent itemset</i> karena nilai minimum <i>support</i> nya terlalu tinggi
	4.1	Sistem gagal melakukan <i>generate rule</i> karena sistem tidak menemukan <i>frequent itemset</i>
	4.2	Sistem gagal membentuk <i>rule</i> karena nilai minimum <i>confidence</i> terlalu tinggi

5. Skenario *use case* Pembentukan paket

Skenario *use case* Pembentukan paket menggambarkan langkah aksi *user* terhadap sistem untuk melakukan filterisasi pada *rule* yang terbentuk dengan memasukkan rentang harga. Untuk lebih jelasnya dapat dilihat pada tabel 3.43 dan 3.44 dibawah :

Tabel 1.43 Requirement A.3

<i>Requirement A.4</i>
Sistem menyediakan menu pembentukan paket untuk seleksi <i>rule</i> yang terbentuk dengan memasukkan rentang harga, dan menampilkan menu apa saja yang dapat dijadikan satu paket sesuai dengan rentang harga yang dimasukan.

Tabel 1.44 Skenario use case Pembentukan paket

<i>Use case name</i>		Pembentukan paket
<i>Related Requirements</i>		A.1, A.2, A.3
<i>Goal in Context</i>		Sistem mendapatkan hasil <i>rule</i> yang sesuai dengan rentang harga yang dimasukkan
<i>Precondition</i>		Data sudah melalui proses asosiasi
<i>Successful End Condition</i>	<i>End</i>	Sistem berhasil menampilkan paket yang direkomendasikan
<i>Failed End Condition</i>		Sistem gagal menampilkan paket yang direkomendasikan
<i>Actors</i>		<i>Manager</i>
<i>Triger</i>		<i>Manager</i> memasukkan rentang harga
<i>Included Cases</i>		<i>Asosiasi</i>
<i>Main flow</i>	<i>Step</i>	<i>Action</i>
	1	Pada menu pembentukan paket, manager memasukkan rentang harga
	2	Sistem menghitung jumlah harga dari setiap <i>rule</i> yang terbentuk
	3	Sistem menampilkan <i>rule</i> yang memenuhi rentang harga

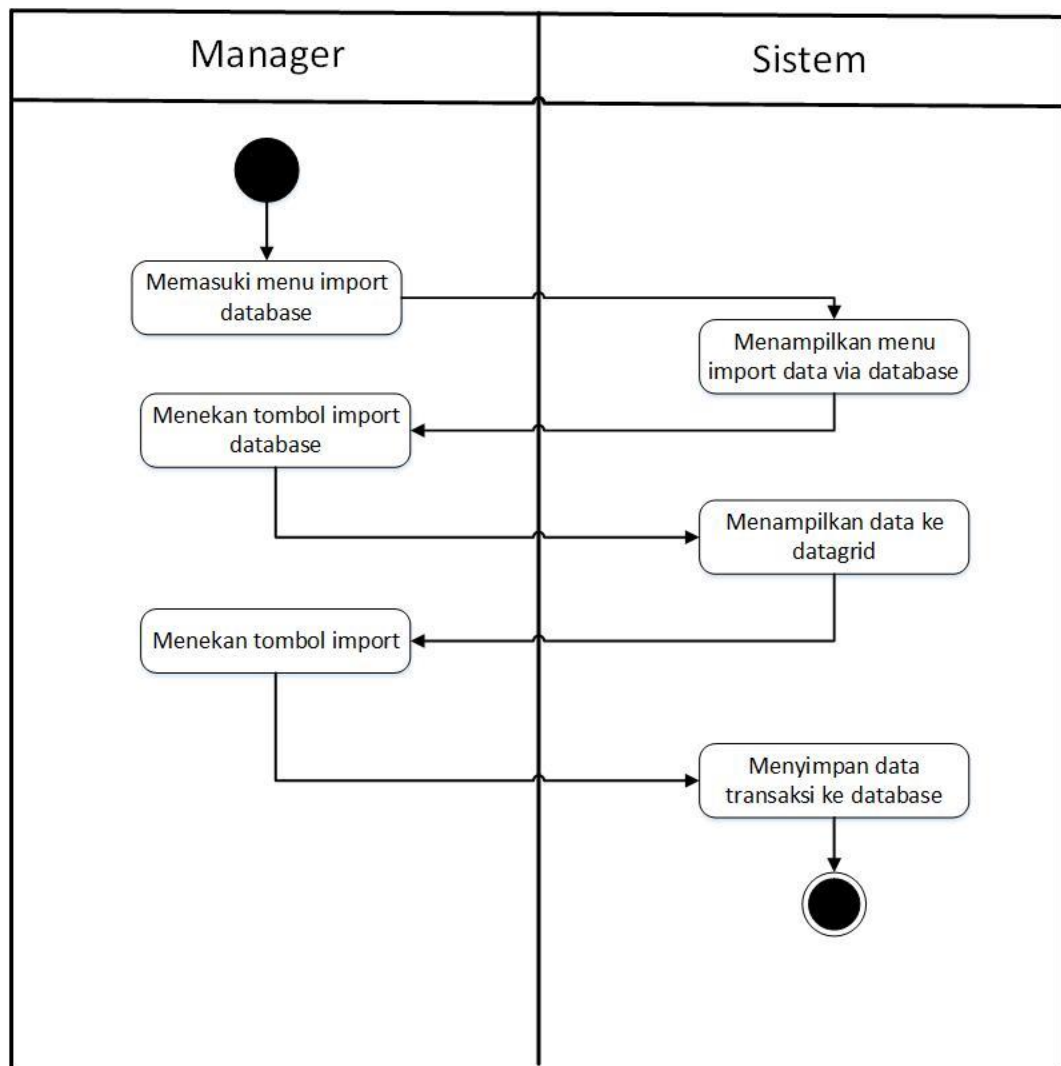
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	1.1	<i>Manager</i> tidak memasukkan rentang harga
	3.1	Sistem tidak mendapatkan <i>rule</i> karena rentang harga yang tidak sesuai

3.1.4.2. *Activity Diagram*

Diagram aktivitas adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem. Diagram aktifitas dari setiap proses dapat dilihat dibawah ini :

1. *Activity Diagram Import Data* dari *database*

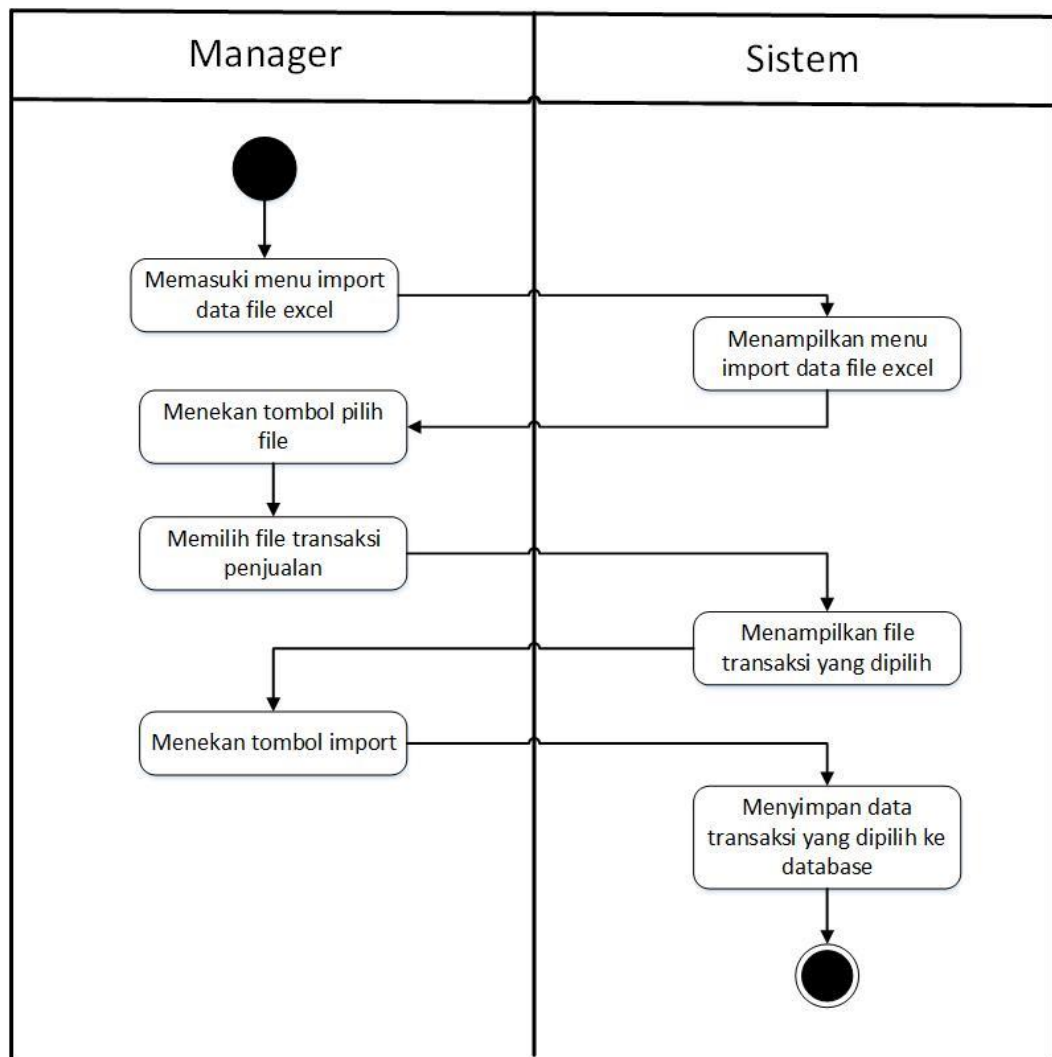
Activity Diagram import data dari *database* menjelaskan aktivitas aktor dan sistem dalam melakukan proses *import data* transaksi langsung dari *database* penjualan kafe Kopi Ciik. *Activity diagram import data* dari *database* diacu pada Gambar 3.37:



Gambar 1.37 Activity diagram import data dari database

2. Activity Diagram Import Data dari file excel

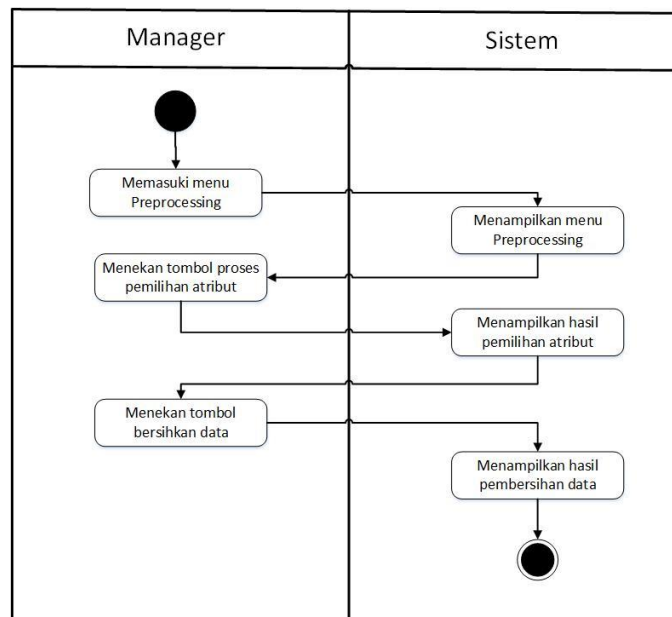
Activity Diagram import data dari file excel menjelaskan aktivitas aktor dan sistem dalam melakukan proses *import data* transaksi dari file berformat Microsoft excel (.xlsx). Activity diagram import data dari file excel diacu pada Gambar 3.38:



Gambar 1.38 Activity diagram import data dari file excel

3. Activity Diagram Preprocessing

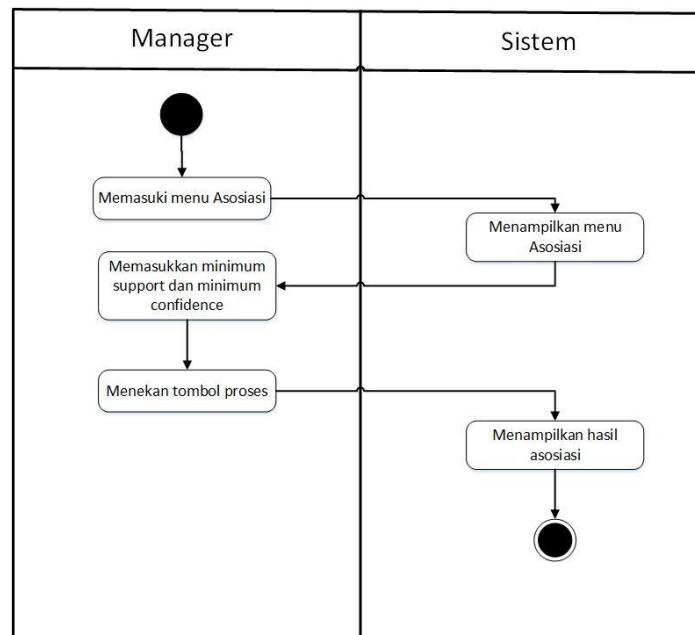
Activity Diagram Preprocessing menjelaskan aktivitas aktor dan sistem dalam melakukan *Preprocessing*. *Activity diagram preprocessing* diacu pada Gambar 3.39:



Gambar 1.39 Activity diagram Preprocessing

4. Activity Diagram Asosiasi

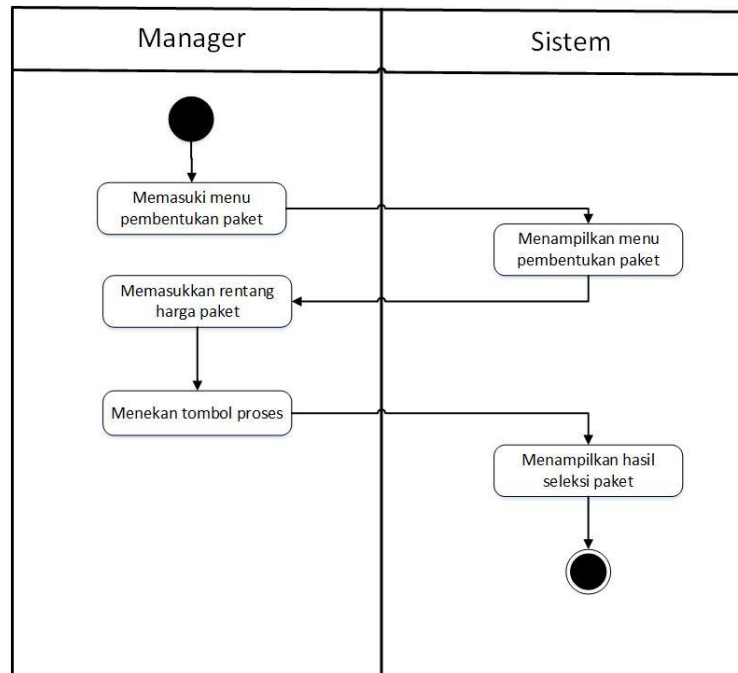
Activity Diagram Asosiasi menjelaskan aktivitas aktor dan sistem dalam melakukan proses asosiasi pada data yang telah melalui tahap *preprocessing*. *Activity Diagram* asosiasi diacu pada Gambar 3.40:



Gambar 1.40 Activity diagram Asosiasi

5. Activity Diagram Pembentukan paket

Activity Diagram Pembentukan paket menjelaskan aktivitas aktor dan sistem dalam melakukan proses pembentukan paket dari *rule* yang telah terbentuk sesuai harga yang telah ditentukan. Activity diagram pembentukan paket diacu pada Gambar 3.41:



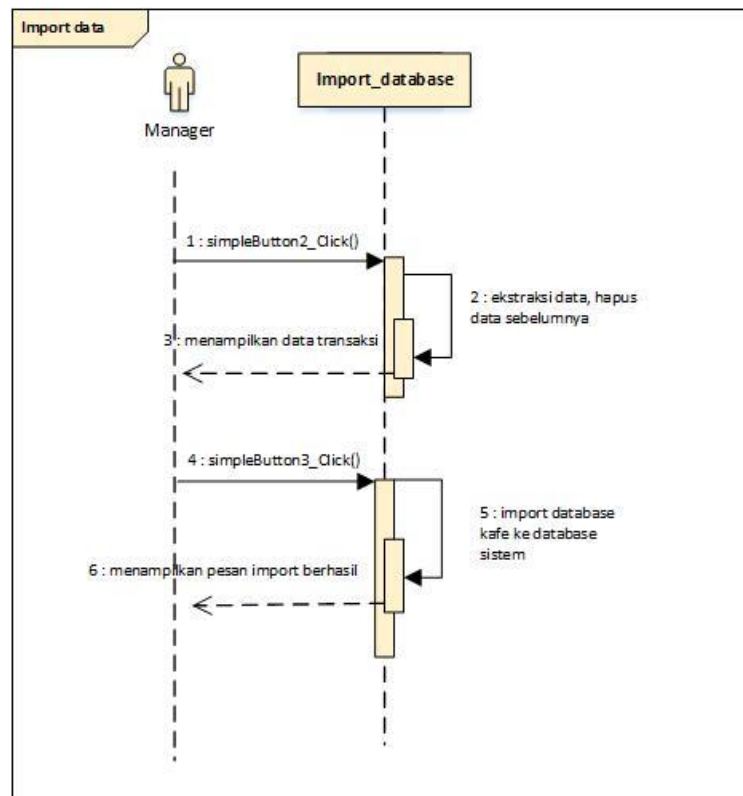
Gambar 1.41 Activity diagram Pembentukan paket

3.1.4.3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

1. Sequence diagram Import data dari database

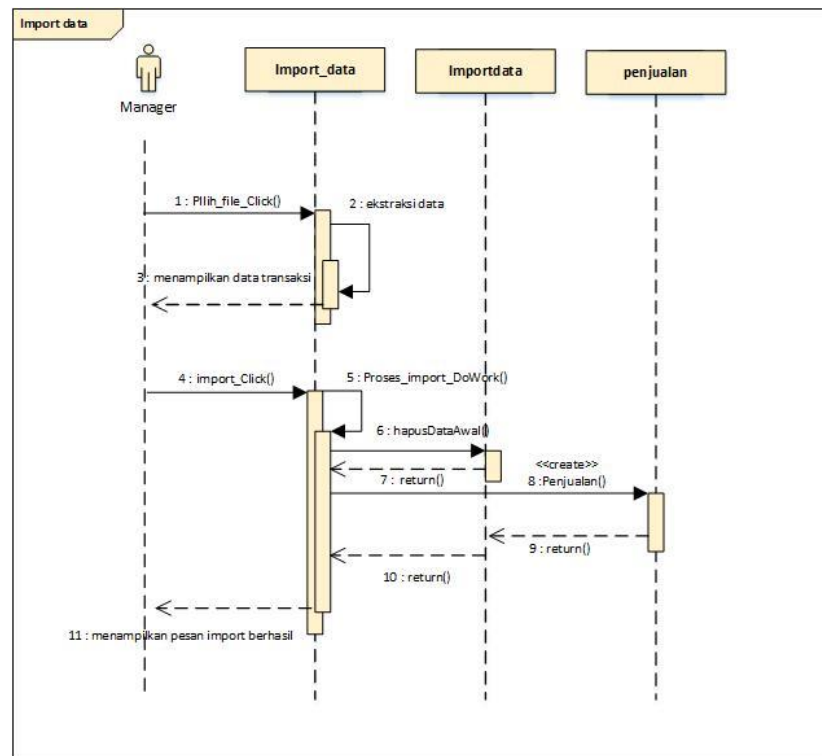
Sequence diagram pada proses *import data* dari *database* dapat dilihat pada gambar 3.42 dibawah ini :



Gambar 1.42 Sequence diagram Import data dari database

2. Sequence diagram Import data dari file excel

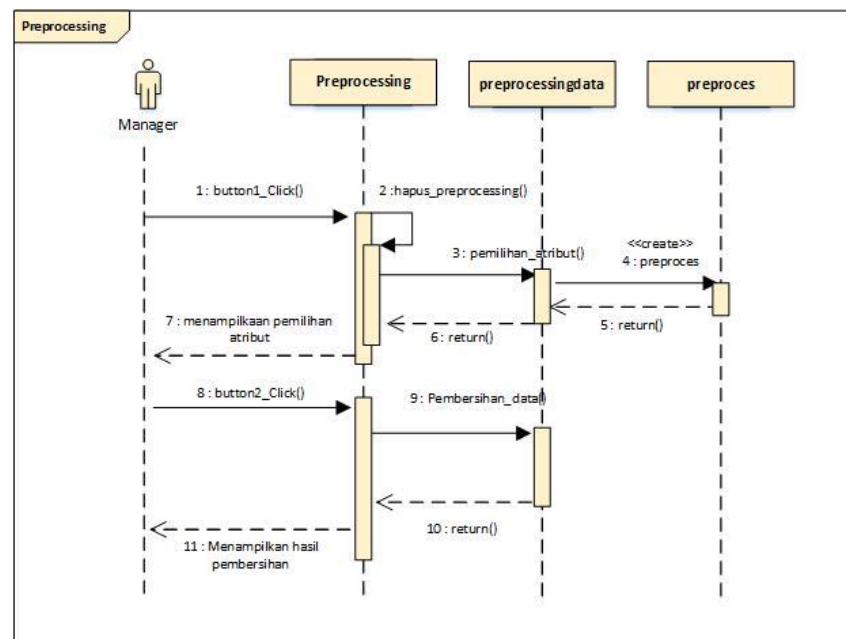
Sequence diagram pada proses *import data* dari file excel dapat dilihat pada gambar 3.43 dibawah ini :



Gambar 1.43 Sequence diagram Import data dari file excel

3. Sequence diagram Preprocessing

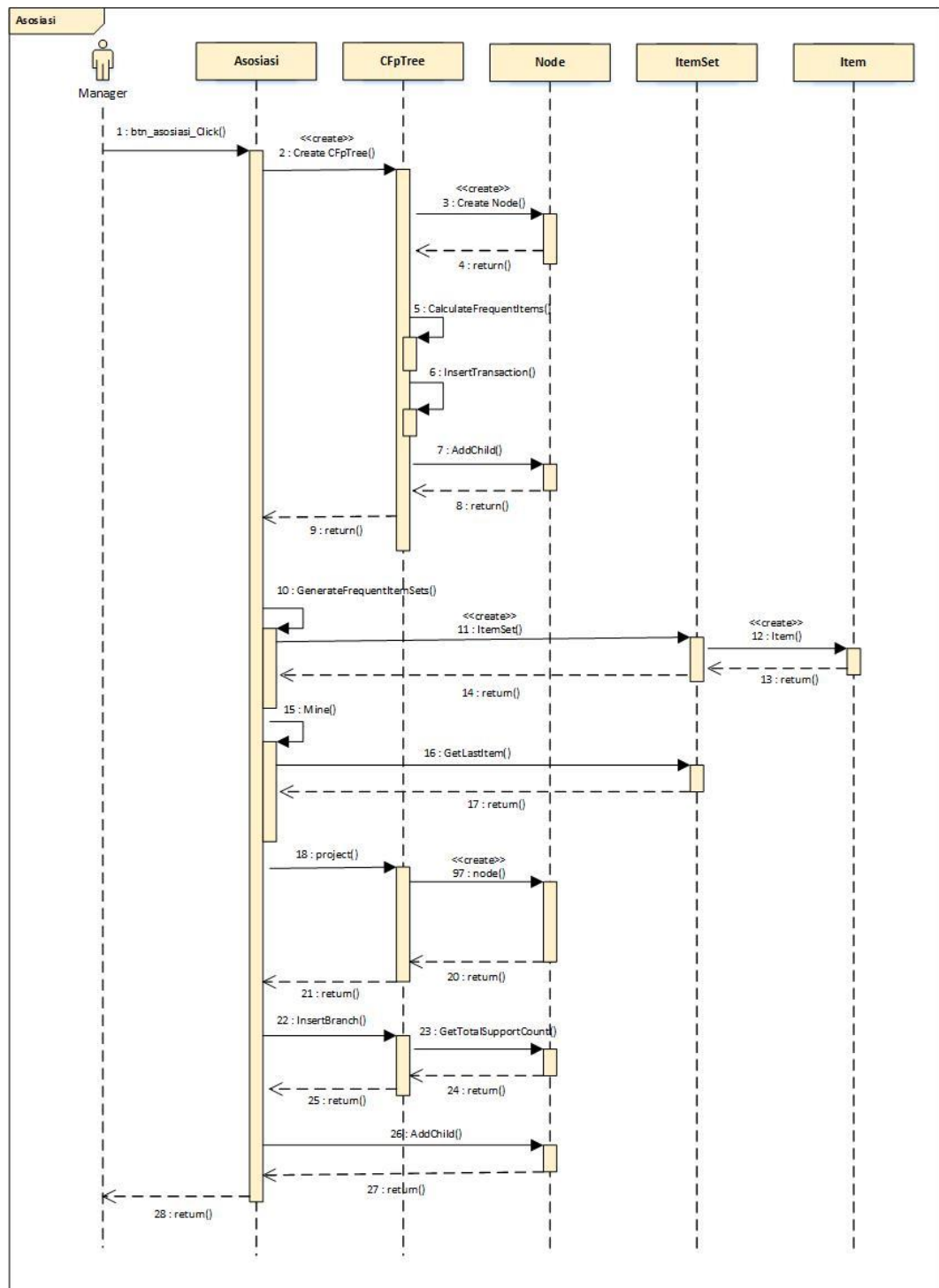
Sequence diagram pada proses *preprocessing* dapat dilihat pada gambar 3.44 dibawah ini :



Gambar 1.44 Sequence diagram Preprocessing

4. Sequence diagram Asosiasi

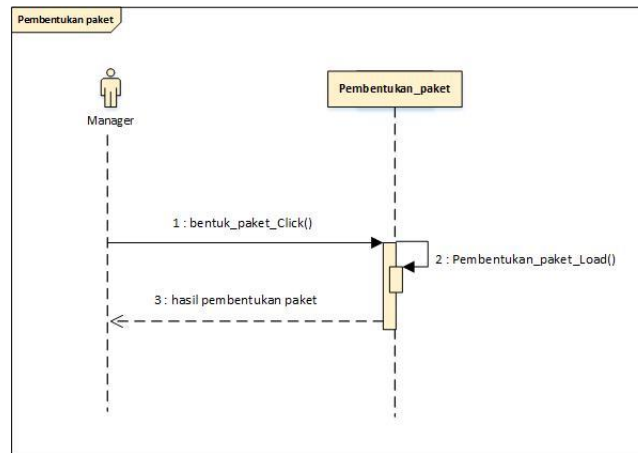
Sequence diagram pada proses asosiasi dapat dilihat pada gambar 3.45 dibawah ini :



Gambar 1.45 Sequence diagram Asosiasi

5. Sequence diagram Pembentukan paket

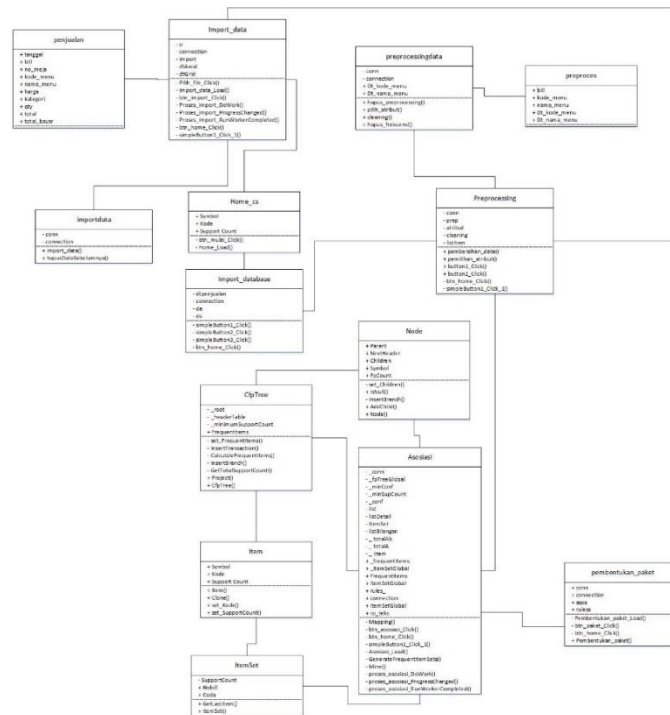
Sequence diagram pada proses pembentukan paket dapat dilihat pada gambar 3.46 dibawah ini :



Gambar 1.46 Sequence diagram Pembentukan paket

3.1.4.4. Class Diagram

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



Gambar 1.47 Class diagram

3.2. Perancangan Sistem

Perancangan sistem merupakan kegiatan dalam perencanaan rancang bangun sistem, mulai dari perancangan data, perancangan antarmuka, perancangan struktur menu, dan perancangan jaringan semantik. Rancangan yang matang, akan menjadikan sistem jauh menjadi lebih baik.

3.2.1. Perancangan Data

Data yang digunakan dalam perancangan sistem ini terdiri dari tabel sebagai berikut :

1. Tabel transaksi
2. Tabel preprocessing
3. Tabel frekuensi
4. Tabel rules

3.2.1.1. Struktur Tabel

1. Tabel transaksi

Tabel transaksi digunakan untuk menyimpan data transaksi hasil *import* dari *file excel* yang digunakan sistem. Struktur pada tabel transaksi dapat dilihat pada pada tabel 3.45 :

Tabel 1.45 Struktur tabel transaksi

Nama <i>field</i>	Tipe data	Keterangan
id	Int	<i>Primary key</i>
tanggal	Date	
Bill	Varchar	
No_meja	Varchar	
Kode_menu	Varchar	
Nama_menu	Varchar	
Kategori	Varchar	
Harga	Float	
Qty	Int	
Total	Float	
Total_bayar	Float	

2. Tabel *preprocessing*

Tabel *preprocessing* digunakan untuk menyimpan hasil dari pemilihan atribut dan pembersihan data yang dilakukan sistem. Struktur pada tabel *preprocessing* dapat dilihat pada tabel 3.46 :

Tabel 1.46 Struktur tabel *preprocessing*

Nama <i>field</i>	Tipe data	Keterangan
id	Int	<i>Foreign key</i> dari tabel transaksi
Bill	Varchar	
Kode_menu	Varchar	
Nama_menu	Varchar	

3. Tabel frekuensi

Tabel frekuensi digunakan untuk menyimpan *support count* dari masing-masing *item*. Struktur pada tabel frekuensi dapat dilihat pada tabel 3.47 :

Tabel 1.47 Struktur tabel frekuensi

Nama <i>field</i>	Tipe data	Keterangan
Id	Int	<i>Foreign key</i> dari tabel transaksi
Kode_menu	Varchar	
Frekuensi	Int	

4. Tabel *rules*

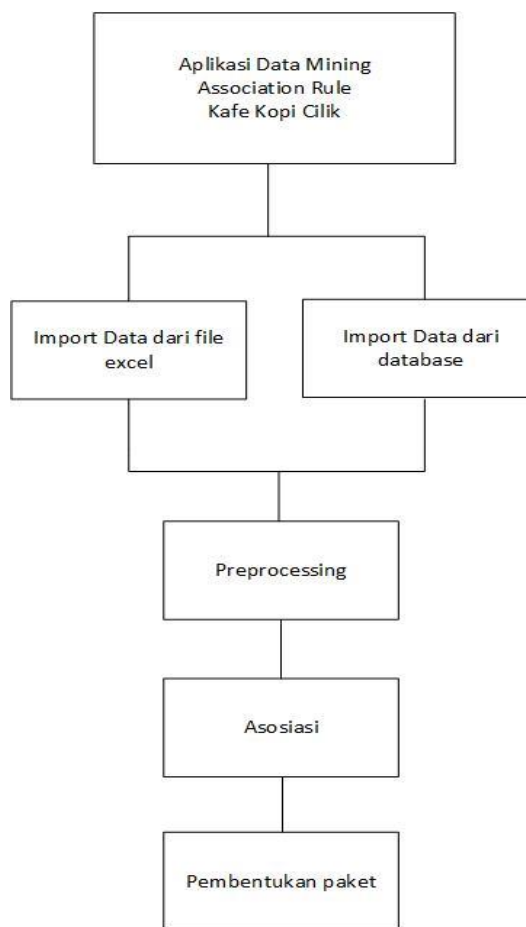
Tabel *rules* digunakan untuk menyimpan *rule* yang dihasilkan dari proses asosiasi. Struktur pada tabel *rules* dapat dilihat pada tabel 3.48 :

Tabel 1.48 Struktur tabel *rules*

Nama <i>field</i>	Tipe data	Keterangan
id	Int	<i>Foreign key</i> dari tabel transaksi
Support	Int	
Confidence	Double	
Rule	Text	

3.2.2. Perancangan Struktur Menu

Dalam pembangunan sistem diperlukan suatu gambaran dari beberapa menu dan keterkaitannya supaya lebih mudah dipahami. Berikut ini pada gambar 3.48 adalah merupakan struktur menu dari sistem yang akan dibangun :



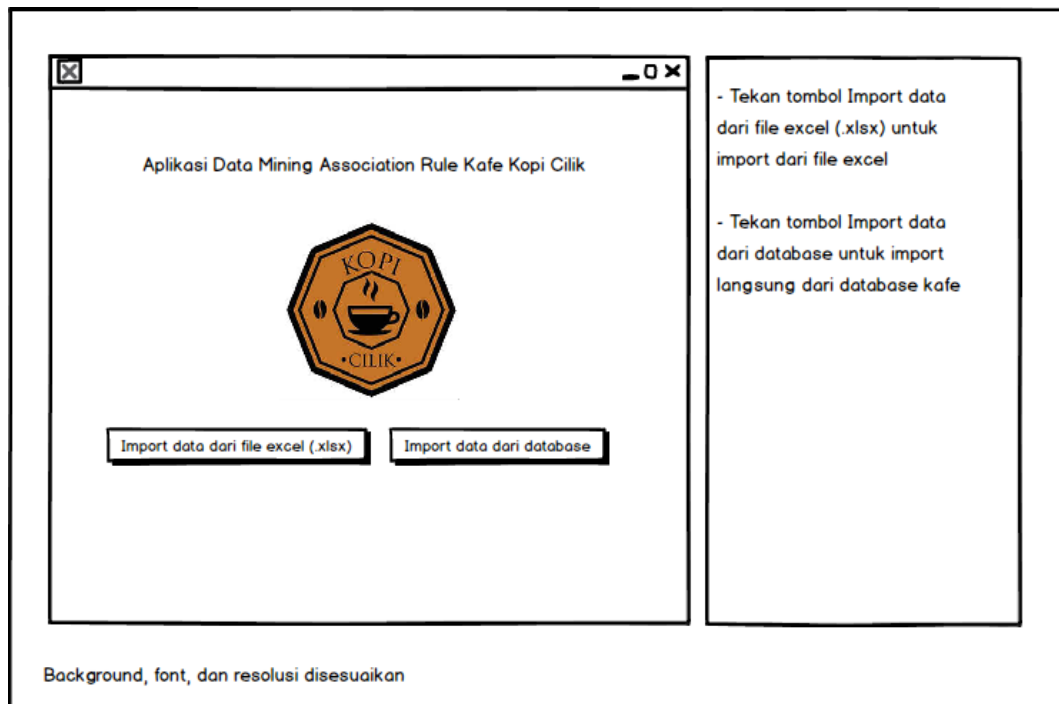
Gambar 1.48 Struktur menu sistem

3.2.3. Perancangan Antarmuka

Perancangan antarmuka untuk sistem yang akan dibangun adalah sebagai berikut :

1. Perancangan antarmuka Halaman utama

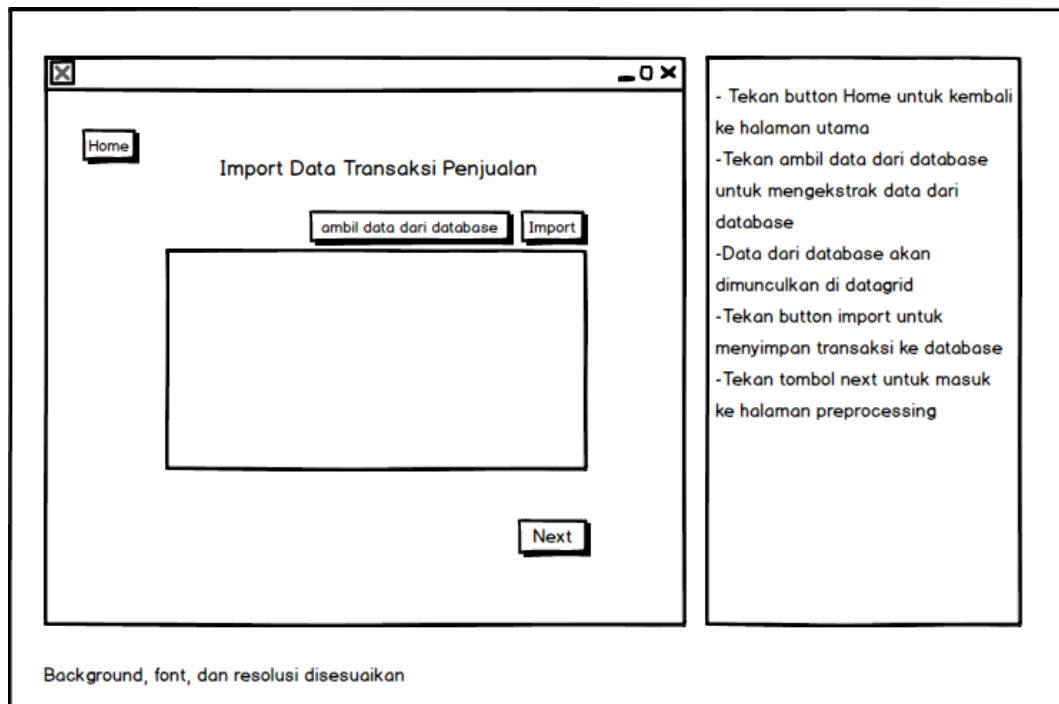
Perancangan antarmuka Halaman utama dapat dilihat pada gambar 3.49 dibawah ini :



Gambar 1.49 Perancangan antarmuka Halaman utama

2. Perancangan antarmuka *Import Data* dari *database*

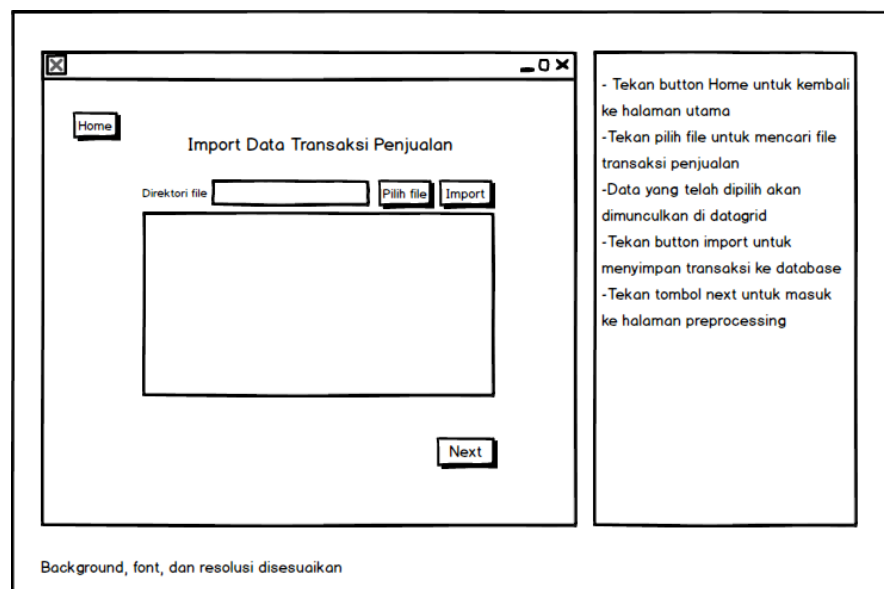
Perancangan antarmuka *Import Data* dari *database* dapat dilihat pada gambar 3.50 dibawah ini :



Gambar 1.50 Perancangan antarmuka *Import Data* dari database

3. Perancangan antarmuka *Import Data* dari file excel

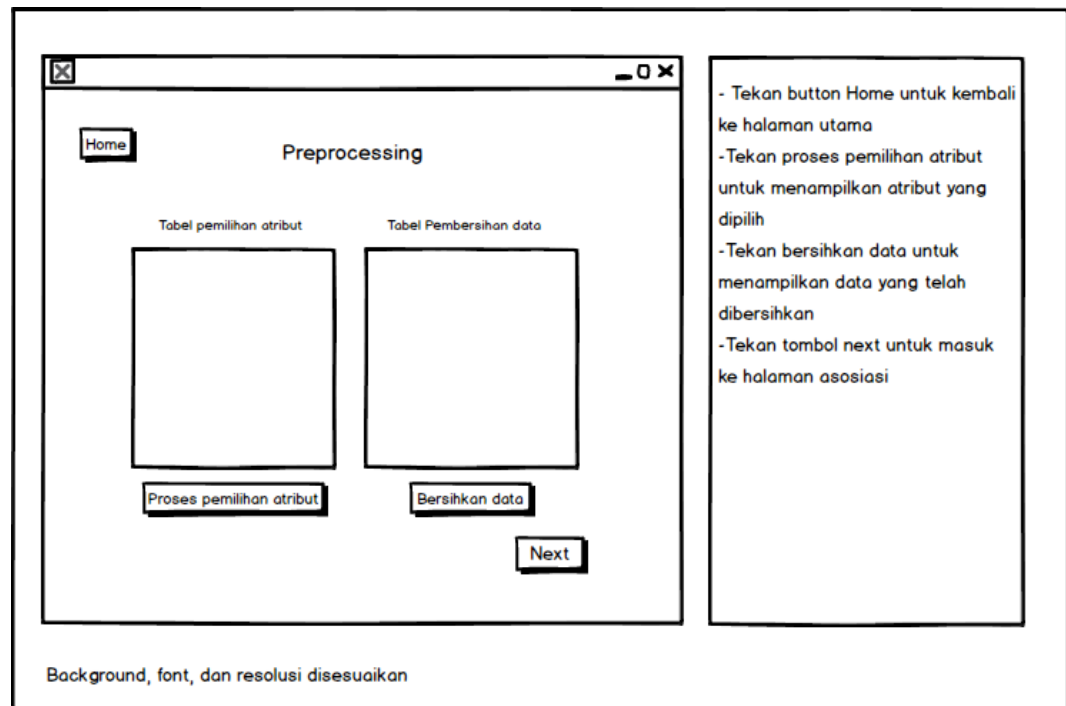
Perancangan antarmuka *Import Data* dari file excel dapat dilihat pada gambar 3.51 dibawah ini :



Gambar 1.51 Perancangan antarmuka *Import Data* dari file excel

4. Perancangan antarmuka *Preprocessing*

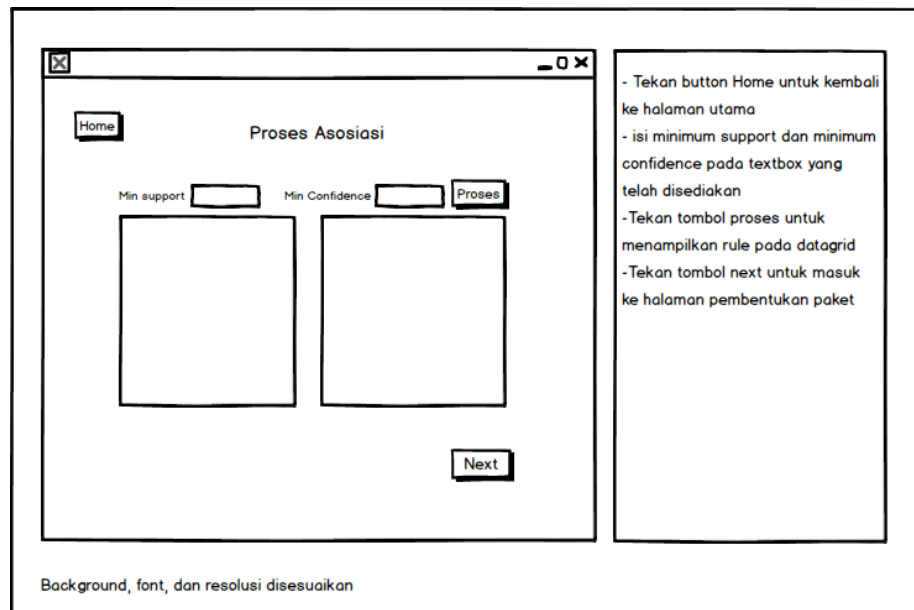
Perancangan antarmuka *Preprocessing* dapat dilihat pada gambar 3.52 dibawah ini :



Gambar 1.52 Perancangan antarmuka *Preprocessing*

5. Perancangan antarmuka Asosiasi

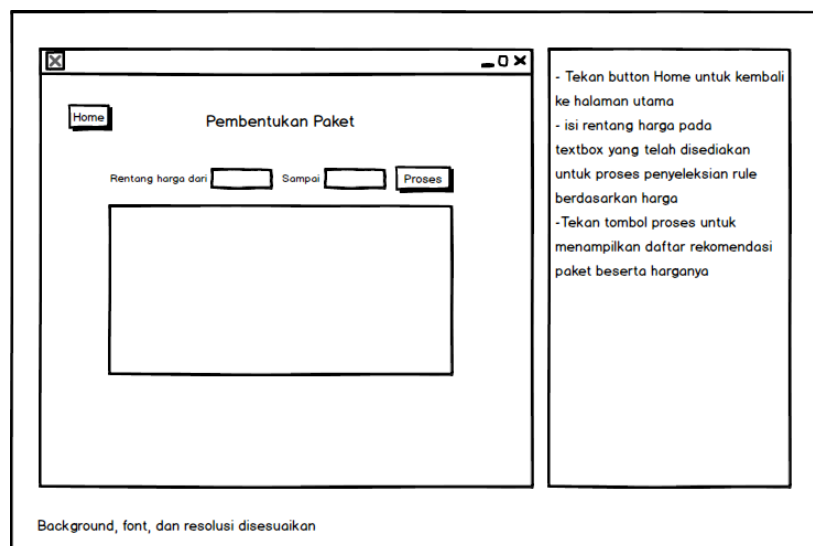
Perancangan antarmuka Asosiasi dapat dilihat pada gambar 3.53 dibawah ini :



Gambar 1.53 Perancangan antarmuka Asosiasi

6. Perancangan antarmuka Pembentukan paket

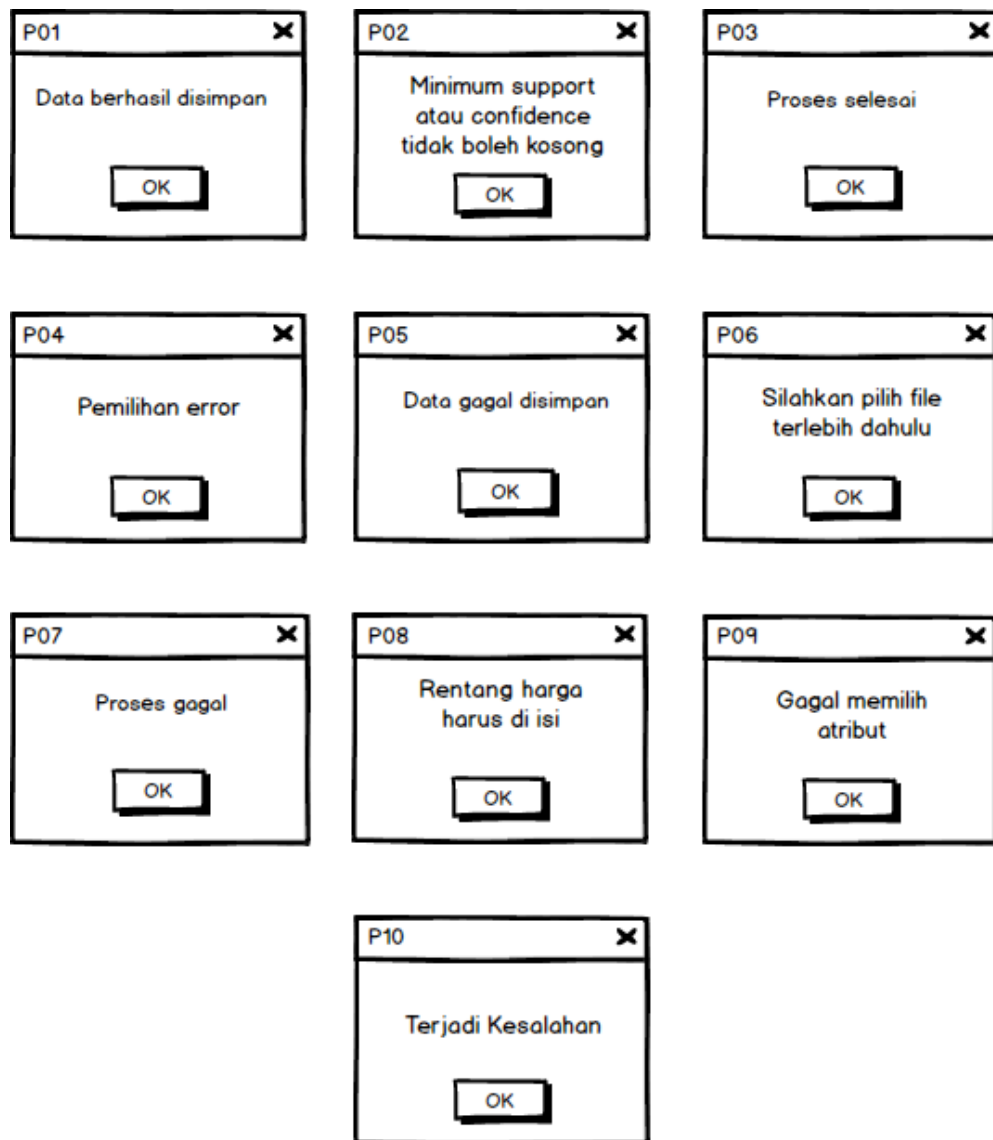
Perancangan antarmuka Pembentukan paket dapat dilihat pada gambar 3.54 dibawah ini :



Gambar 1.54 Perancangan antarmuka Pembentukan paket

3.2.4. Perancangan Pesan

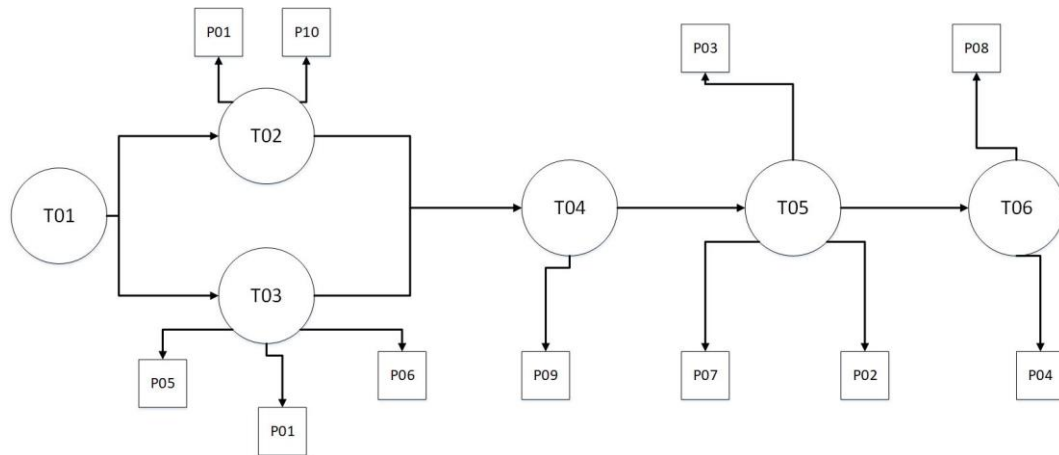
Adapun perancangan pesan pada sistem ini dapat dilihat pada gambar 3.55 dibawah :



Gambar 1.55 Perancangan pesan sistem

3.2.5. Perancangan Jaringan Semantik

Adapun perancangan jaringan semantik pada sistem ini dapat dilihat pada gambar 3.56 dibawah :



Gambar 1.56 Perancangan Jaringan Semantik

