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 Penerapan Metode Crisp-DM
3. Analisis Prosedur Penentuan Paket Produk Kosmetik
4. Analisis Spesifikasi Kebutuhan Perangkat Lunak

3.1.1 Analisis Masalah

Berdasarkan hasil pengamatan, dapat disimpulkan bahwa permasalahan yang ada di toko yaitu belum adanya informasi penentuan paket produk kosmetik yang sesuai dengan minat konsumen yang akan ditawarkan.

3.1.2 Analisis Penerapan Metode Crisp-DM

3.1.2.1 Pemahaman Bisnis

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

1. Penentuan Tujuan Bisnis

Menjual produk dalam bentuk paket untuk memenuhi permintaan konsumen dan menghasilkan keuntungan untuk perusahaan.

2. Menilai situasi

Menilai situasi merupakan tahapan untuk menjelaskan sumber daya yang tersedia di Makmur Jaya Kosmetik. Berikut Merupakan sumber daya yang ada pada Makmur Jaya Kosmetik

1. Sumber Daya Perangkat Keras
    1.1 Terdapat Alat penunjang transaksi yaitu, printer thermal dan cashdrawer (laci Kasir)
    1.2 Komputer
2. Sumber daya data
    2.1 Sumber daya data yang di Makmur Jaya Kosmetik yaitu data transaksi penjualan.
    2.2 Sumber daya data ini bersifat internal yaitu tersimpan di database Makmur Jaya Kosmetik
3. Sumber data personil
    3.1 Pegawai kasir yang bertugas melakukan transaksi pembeli , mencetak nota dan membuat laporan penjualan yang akan di serahkan ke manager.
4. Kebutuhan
    4.1 Data yang digunakan dalam penelitian ini adalah data transaksi penjualan.
5. Asumsi
    5.1 Data transaksi yang digunakan merupakan hasil import dari aplikasi kasir di Makmur Jaya Kosmetik dengan format database (.sql)
    5.2 Data transaksi yang digunakan periode bulan Januari 2016
6. Batasan
    6.1 Hasil dari data mining ini berupa rule sebagai acuan untuk membuat paket.

3. Penentuan Sasaran *Data Mining*

Mengetahui keterkaitan produk yang sering dibeli oleh kosumen sehingga bisa dijadikan acuan dalam penentuan paket produk.

3.1.2.2 Pemahaman Data

Tahapan pemahaman data dimulai dengan pengumpulan data awal dan hasil kegiatan untuk mengidentifikasi masalah data, untuk mendetekdi *subset* menarik untuk membetuk hipotesis untuk informasi yang tersembunyi.

1. Mengumpulkan Data Awal

Sumber yang didapat adalah data transaksi di Makmur Jaya Kosmetik. Yang dapat dilihat di Gambar 3.1.



Makmur Jaya Kosmetik

Jl.Jend A.Yani No.62, Garut

Tlp. : (0262) 233624

No Faktur:16010001

Tanggal:1/1/2016

Kode	Nama Produk	
Wfs60	Wardah Facial Scrub 60 ml	
	1x Rp.39.000	Rp.39.000
PCSAC40	PONDS CS ACNE C 40	
	1x Rp.22.500	Rp.22.500
CHPWGUV40	CITRA HAZ SPOTLES	
	1x Rp.10.500	Rp.10.500
LTMC120	La Tulip Milk Cl 120 ml	
	1x Rp.23.000	Rp.23.000
KS35	KAPAS SARIAYU 35	
	1x Rp.5.000	Rp5.000
PAR	Pensil Alis Revlon	
	1x Rp.5.000	Rp5.000
PSHPAD	PANTENE SHP ANTI	
	1x Rp.22.000	Rp.22.000
WLDCS30	Wardah Lig Day 1 30 gr	
	1x Rp.24.000	Rp.24.000
	Total	Rp.151.000

Produk yang sudah dibeli tidak dapat dikembalikan atau ditukar dengan barang yang lain

Gambar 3. 1 Data Transaksi Makmur Jaya Kosmetik

Data yang ada saat ini merupakan bentuk tidak normal, untuk memudahkan proses analisis dalam data *mining*maka dilakukan proses normalisasi. Berikut adalah tahapan proses normalisasi:

a. Tahap 1: Bentuk *unnormalize*

Tabel 3. 1 Bentuk Unnormalize Data Transaksi

Nama toko	Alamat toko	Tlp toko	No Faktur	Tang gal	Kode	Nama Produk	Qty	Harga	Jumla h	Total Harga
Makmur Jaya Kosmetik	Jl. Jend A.Yani No.62, Garut	(0262) 233624	16010001	01/01 /2016	WFS60	Wardah Facial Scrub 60 ml	1	39000	39000	5541500
					PCSAC40	PONDS CS ACNE C 40	1	22500	22500	
					CHPWGUV40	CITRA HAZ SPOTLES WHITE GLOW UV 40 G	1	10500	10500	
					LTMC120	La Tulip Milk Cleanser 120 ml New	1	23000	23000	
					KS35	KAPAS SARIAYU 35	1	5000	5000	
					PAR	Pensil Alis Revlon	1	5000	5000	
					PSHPAD170	PANTENE SHP ANTI DANRUFF 170 ML ATLAS	1	22000	22000	
					WLDCS30S1	Wardah Lig Day Cream Step 1 30 gr	1	24000	24000	

b. Tahap 2: Bentuk Normal Ke-1

Tabel 3. 2 Bentuk Norma ke-1

No Faktur	Tanggal	Kode	Nama Produk	Qty	Harga	Jumla h	Total Harga
16010001	01/01/2016	WFS60	Wardah Facial Scrub 60 ml	1	39000	39000	5541500
16010001	01/01/2016	PCSAC40	PONDS CS ACNE C 40	1	22500	22500	5541500
16010001	01/01/2016	CHPWGUV40	CITRA HAZ SPOTLES WHITE GLOW UV 40 G	1	10500	10500	5541500
16010001	01/01/2016	LTMC120	La Tulip Milk Cleanser 120 ml New	1	23000	23000	5541500

16010001	01/01/2016	KS35	KAPAS SARIAYU 35	1	5000	5000	5541500
16010001	01/01/2016	PAR	Pensil Alis Revlon	1	5000	5000	5541500
16010001	01/01/2016	PSHPAD170	PANTENE SHP ANTI DANRUFF 170 ML ATLAS	1	22000	22000	5541500
16010001	01/01/2016	WLDCS30S1	Wardah Lig Day Cream Step 1 30 gr	1	24000	24000	5541500

2. Mendeskripsikan Data

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

Tabel 3. 3 Struktur Data Transaksi Penjualan

Dokumen	**Keterangan**		
Detail Data Transaksi	Deskripsi	Data ini berisi mengenai data transaksi yang ada di toko	
	Format	Microsoft Excel (.xlsx)	
	Atribut	No Faktur	Nomor struk
		Tanggal	Tanggal pencetakan struk
		Kode	Kode barang yang dibeli
		Nama Produk	Nama barang yang dibeli
		Qty	Jumlah barang yang dibeli
		Harga	Harga barang yang dibeli
		Jumlah	Harga Barang dikalikan dengan Qty
		Total Harga	Total harga dari barang yang dibeli

3.1.2.2.3 Verify Data Quality (Verifikasi Kualitas Data)

Dari data transaksi yang didapat terdapat noise, *noise* yaitu data transaksi yang hanya memiliki 1 barang dalam sekali transaksi sehingga data tersebut perlu dihilangkan dikarenakan data tersebut akan mempengaruhi proses *mining* kedepanya. Maka dari itu data – data yang memiliki *noise* akan dilakukan pembersihan data pada tahap berikutnya.

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. Ekstrasi Data

Ekstraksi data dapat diartikan sebagai proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data ke tingkat selanjutnya ataupun untuk menyimpan data hasil ekstrak tersebut. Dalam penelitian ini, data yang berasal dari *flat file* berformat *microsoft excel* (.xlsx) di ekstrak kedalam format *sql*, kemudian disimpan kedalam sebuah *database* agar memudahkan dalam proses pengolahan data. Berikut merupakan contoh sampel data transaksi 1 bulan periode bulan Januari sebanyak 30 transaksi yang terdapat di toko , dimana atribut-atribut yang terdapat dalam data tersebut antara lain No, No Faktur, Tanggal, Kode Barang, Nama Barang, Varian, Harga Barang, Qty, Jumlah, dan Total Harga.

2. Pemilihan Atribut (Select Data)

Proses pemilihan atribut atau *selection data* adalah proses dimana atribut data akan dipilih dan diproses sesuai dengan kebutuhan *data mining*. Sebelum melakukan proses *cleaning* atau pembersihan data akan lebih efisien jika melakukan proses *selection* atau pemilihan atribut ini terlebih dahulu. Karena dari data transaksi yang sebanyak ribuan record ini memiliki 10 atribut sedangkan yang dibutuhkan untuk *data mining* hanya 2 atribut. Dengan menyeleksi atribut yang tidak dibutuhkan akan memudahkan pada proses pembersihan data nanti, sehingga pada saat pembersihan data tidak akan memperberat memori karena program harus membaca atribut yang

tidak perlu. Dalam penelitian ini, 2 atribut yang akan digunakan adalah atribut No Faktur dan Kode. K edua atribut ini digunakan untuk memenuhi tujuan awal dimana akan dicari pola pembelian konsumen berdasarkan produk yang dibeli. Seperti atribut No Faktur digunakan untuk membedakan satu transaksi dengan transaksi lainnya, dan atribut Kode digunakan untuk mengetahui produk apa saja yang dibeli dalam satu transaksi.

NO	NoFaktur	Tanggal	KodeProduk	NamaProduk	Qty	Harga	Jumlah	Total
1	16010001	01/01/2016	WF560	Wardah Facial Scrub 60 ml	1	39000	39000	5541500
2	16010001	01/01/2016	PCSAC40	PONDS CS ACNE C 40	1	22500	22500	5541500
3	16010001	01/01/2016	CHPWGUV40	CITRA HAZ SPOTLES WHITE GLOW UV 40 G	1	10500	10500	5541500
4	16010001	01/01/2016	LTMC120	La Tulip Milk Cleanser 120 ml New	1	23000	23000	5541500
5	16010001	01/01/2016	KS35	KAPAS SARIAYU 35	1	5000	5000	5541500
6	16010001	01/01/2016	PAR	Pensil Alis Revlon	1	5000	5000	5541500
7	16010001	01/01/2016	PSHPAD170	PANTENE SHP ANTI DANRUFF 170 ML ATLAS	1	22000	22000	5541500
8	16010001	01/01/2016	WLDCS30S1	Wardah Lig Day Cream Step 1 30 gr	1	24000	24000	5541500
9	16010002	01/01/2016	WF560	Wardah Facial Scrub 60 ml	1	39000	39000	5541500
10	16010002	01/01/2016	PCSAC40	PONDS CS ACNE C 40	1	22500	22500	5541500
11	16010002	01/01/2016	CHPWGUV40	CITRA HAZ SPOTLES WHITE GLOW UV 40 G	1	10500	10500	5541500
12	16010002	01/01/2016	MSDHM500	MAKARIZO SALON DAILY HAIR MASK 500gr NEW	1	75000	75000	5541500
13	16010002	01/01/2016	PFLC25	PONDS FAIR LOVELY CREAM 25 G	1	11500	11500	5541500
14	16010002	01/01/2016	KS35	KAPAS SARIAYU 35	1	5000	5000	5541500
15	16010002	01/01/2016	PELB	Pixy Eye Liner Black	1	28000	28000	5541500
16	16010002	01/01/2016	PSHPAD170	PANTENE SHP ANTI DANRUFF 170 ML ATLAS	1	22000	22000	5541500

Tabel 3. 4 Pemilihan Atriburt

NoFaktur	Kode
16010001	WFS60
16010001	PCSAC40
16010001	CHPWGUV40
16010001	LTMC120
16010001	KS35
16010001	PAR
16010001	PSHPAD170
16010001	WLDCS30S1
16010002	WFS60
16010002	PCSAC40
16010002	CHPWGUV40
16010002	MSDHM500
16010002	PFLC25
16010002	KS35
16010002	PELB
16010002	PSHPAD170
16010003	WFS60

NoFaktur	Kode
16010014	PELB
16010014	WFSNDS
16010015	WLCF60
16010015	PCSAC40
16010015	MSDHM500
16010015	PFLC25
16010015	PCSACWFF50
16010016	CHPWGUV40
16010016	LTMC120
16010016	WLNC20S2
16010016	RAWL10
16010016	ELDW
16010016	JMS
16010016	PLDS35
16010017	WFS60
16010017	PCSAC40
16010017	CHPWGUV40

16010003	WLCF60
16010003	CHPWGUV40
16010003	MSDHM500
16010003	KS35
16010003	PELB
16010003	WLDCS30S1
16010003	WLNC20S2
16010003	WFSNDS
16010003	WELFSP20
16010004	WLCF60
16010004	PCSAC40
16010004	CHPWGUV40
16010004	PELB
16010004	PSHPAD170
16010005	WFS60
16010005	PCSAC40
16010005	MSDHM500
16010005	LTMC120
16010005	PAR
16010005	PELB
16010005	WLDCS30S1
16010006	WLCF60
16010007	PCSAC40
16010008	WFS60
16010008	CHPWGUV40
16010008	MSDHM500
16010008	KS35
16010008	PELB
16010008	PSHPAD170
16010009	WLCF60
16010009	CHPWGUV40
16010009	KS35
16010009	PAR
16010009	PELB
16010009	WLDCS30S1
16010009	WFSNDS
16010009	SSHP170
16010009	MKP120

16010017	PCSACWFF50
16010017	PAR
16010017	WLBBCL15
16010017	SSHP170
16010018	WFS60
16010018	MSDHM500
16010018	LTMC120
16010018	PFLC25
16010018	MKP120
16010018	ELDW
16010019	WLCF60
16010019	PCSACWFF50
16010019	RAWL10
16010019	HICAT60
16010020	WFS60
16010020	WLCF60
16010020	PFLC25
16010020	PAR
16010020	EHVITR
16010021	WLCF60
16010021	SSHP170
16010022	HICAT60
16010023	WFS60
16010023	PCSAC40
16010023	MSDHM500
16010023	LTMC120
16010023	PFLC25
16010024	WFS60
16010024	PCSACWFF50
16010024	PAR
16010024	WLBBCL15
16010024	EHVITR
16010025	WLCF60
16010025	CHPWGUV40
16010025	PFLC25
16010025	RAWL10
16010025	HICAT60
16010025	MKP120

16010010	WFS60
16010010	PCSAC40
16010010	CHPWGUV40
16010010	MSDHM500
16010010	LTMC120
16010010	KS35
16010010	PELB
16010010	PSHPAD170
16010011	WLCF60
16010011	LTMC120
16010011	KS35
16010011	PSHPAD170
16010011	WLDCS30S1
16010011	WLNC20S2
16010011	WFSNDS
16010011	SSHP170
16010012	WFS60
16010012	WLCF60
16010012	CHPWGUV40
16010012	LTMC120
16010012	PAR
16010012	WLDCS30S1
16010013	WLCF60
16010013	MSDHM500
16010013	KS35
16010013	PSHPAD170
16010013	WLNC20S2
16010013	WFSNDS
16010013	HICAT60
16010014	WFS60
16010014	PCSAC40
16010014	CHPWGUV40
16010014	LTMC120
16010014	KS35
16010014	PAR

16010026	WFS60
16010026	PCSAC40
16010026	MSDHM500
16010026	PCSACWFF50
16010026	WLNC20S2
16010026	WLBBCL15
16010026	SSHP170
16010026	EHVITR
16010027	WFS60
16010027	PCSAC40
16010027	LTMC120
16010027	PFLC25
16010027	PCSACWFF50
16010027	WLBBCL15
16010027	WELFSP20
16010028	WFS60
16010028	WLCF60
16010028	PFLC25
16010028	PCSACWFF50
16010028	RAWL10
16010028	WELFSP20
16010029	WLCF60
16010029	PCSAC40
16010029	MSDHM500
16010029	PFLC25
16010029	PCSACWFF50
16010029	EHVITR
16010030	WLCF60
16010030	PCSAC40
16010030	CHPWGUV40
16010030	MSDHM500
16010030	PFLC25
16010030	PCSACWFF50
16010030	WLNC20S2
16010030	WLBBCL15

3. Pembersihan Data

Proses pembersihan data atau *cleaning data* adalah proses menghilangkan data tidak relevan atau inkosisten dan proses menghilangkan *noise*. *Noise* disini yaitu data transaksi yang hanya memiliki 1 produk dalam sekali pembelian. Dalam data transaksi ini akan dilakukan pengeleminasian terhadap transaksi yang memiliki jumlah produk kurang dari 2 produk dalam satu kali transaksinya, karena syarat ini diperlukan dalam *Association Rules* untuk melihat keterhubungan antar 2 produk atau lebih. Setelah melakukan proses pembersihan data, dari data transaksi yang awalnya sebanyak 33 transaksi di *cleaning* menjadi 27 transaksi.

Sebelum Pembersihan Data

NoFaktur	Kode
16010005	WFS60
16010005	PCSAC40
16010005	MSDHM500
16010005	LTMC120
16010005	PAR
16010005	PELB
16010005	WLDCS30S1
16010006	WLCF60
16010007	PCSAC40
16010008	WFS60
16010008	CHPWGUV40
16010008	MSDHM500
16010008	KS35
16010008	PELB
16010008	PSHPAD170

Setelah Pembersihan Dara

NoFaktur	Kode
16010005	WFS60
16010005	PCSAC40
16010005	MSDHM500
16010005	LTMC120
16010005	PAR
16010005	PELB
16010005	WLDCS30S1
16010008	WFS60
16010008	CHPWGUV40
16010008	MSDHM500
16010008	KS35
16010008	PELB
16010008	PSHPAD170

Tabel 3. 5 Hasil Pembersihan Data

NoFaktur	Kode
16010001	WFS60
16010001	PCSAC40
16010001	CHPWGUV40
16010001	LTMC120
16010001	KS35

NoFaktur	Kode
16010014	WFSNDS
16010015	WLCF60
16010015	PCSAC40
16010015	MSDHM500
16010015	PFLC25

16010001	PAR
16010001	PSHPAD170
16010001	WLDCS30S1
16010002	WFS60
16010002	PCSAC40
16010002	CHPWGUV40
16010002	MSDHM500
16010002	PFLC25
16010002	KS35
16010002	PELB
16010002	PSHPAD170
16010003	WFS60
16010003	WLCF60
16010003	CHPWGUV40
16010003	MSDHM500
16010003	KS35
16010003	PELB
16010003	WLDCS30S1
16010003	WLNC20S2
16010003	WFSNDS
16010003	WELFSP20
16010004	WLCF60
16010004	PCSAC40
16010004	CHPWGUV40
16010004	PELB
16010004	PSHPAD170
16010005	WFS60
16010005	PCSAC40
16010005	MSDHM500
16010005	LTMC120
16010005	PAR
16010005	PELB
16010005	WLDCS30S1
16010008	WFS60
16010008	CHPWGUV40
16010008	MSDHM500
16010008	KS35
16010008	PELB
16010015	PCSACWFF50
16010016	CHPWGUV40
16010016	LTMC120
16010016	WLNC20S2
16010016	RAWL10
16010016	ELDW
16010016	JMS
16010016	PLDS35
16010017	WFS60
16010017	PCSAC40
16010017	CHPWGUV40
16010017	PCSACWFF50
16010017	PAR
16010017	WLBBCL15
16010017	SSHP170
16010018	WFS60
16010018	MSDHM500
16010018	LTMC120
16010018	PFLC25
16010018	MKP120
16010018	ELDW
16010019	WLCF60
16010019	PCSACWFF50
16010019	RAWL10
16010019	HICAT60
16010020	WFS60
16010020	WLCF60
16010020	PFLC25
16010020	PAR
16010020	EHVITR
16010021	WLCF60
16010021	SSHP170
16010023	WFS60
16010023	PCSAC40
16010023	MSDHM500
16010023	LTMC120
16010023	PFLC25
16010024	WFS60

16010008	PSHPAD170
16010009	WLCF60
16010009	CHPWGUV40
16010009	KS35
16010009	PAR
16010009	PELB
16010009	WLDCS30S1
16010009	WFSNDS
16010009	SSHP170
16010009	MKP120
16010010	WFS60
16010010	PCSAC40
16010010	CHPWGUV40
16010010	MSDHM500
16010010	LTMC120
16010010	KS35
16010010	PELB
16010010	PSHPAD170
16010011	WLCF60
16010011	LTMC120
16010011	KS35
16010011	PSHPAD170
16010011	WLDCS30S1
16010011	WLNC20S2
16010011	WFSNDS
16010011	SSHP170
16010012	WFS60
16010012	WLCF60
16010012	CHPWGUV40
16010012	LTMC120
16010012	PAR
16010012	WLDCS30S1
16010013	WLCF60
16010013	MSDHM500
16010013	KS35
16010013	PSHPAD170
16010013	WLNC20S2
16010013	WFSNDS
16010024	PCSACWFF50
16010024	PAR
16010024	WLBBCL15
16010024	EHVITR
16010025	WLCF60
16010025	CHPWGUV40
16010025	PFLC25
16010025	RAWL10
16010025	HICAT60
16010025	MKP120
16010026	WFS60
16010026	PCSAC40
16010026	MSDHM500
16010026	PCSACWFF50
16010026	WLNC20S2
16010026	WLBBCL15
16010026	SSHP170
16010026	EHVITR
16010027	WFS60
16010027	PCSAC40
16010027	LTMC120
16010027	PFLC25
16010027	PCSACWFF50
16010027	WLBBCL15
16010027	WELFSP20
16010028	WFS60
16010028	WLCF60
16010028	PFLC25
16010028	PCSACWFF50
16010028	RAWL10
16010028	WELFSP20
16010029	WLCF60
16010029	PCSAC40
16010029	MSDHM500
16010029	PFLC25
16010029	PCSACWFF50
16010029	EHVITR
16010030	WLCF60

16010013	HICAT60
16010014	WFS60
16010014	PCSAC40
16010014	CHPWGUV40
16010014	LTMC120
16010014	KS35
16010014	PAR
16010014	PELB

16010030	PCSAC40
16010030	CHPWGUV40
16010030	MSDHM500
16010030	PFLC25
16010030	PCSACWFF50
16010030	WLNC20S2
16010030	WLBBCL15

3.1.2.4 Modeling

Pemodelan merupakan tahap untuk mebuat model atau desain sistem yang akan dibangun. Data yang digunakan untuk pengolahan data adalah data yang tertera pada tabel 3.3

1. Select Modeling Technique (Memilih Teknik Pemodelan)

Dengan menggunakan metode *Data Mining*, data transaksi akan diolah dengan aturan asosiasi untuk menemukan pola-pola pembelian barang yang sering dibeli oleh konsumen. Didalam tahap ini metode yang digunakan adalah *Assosiacion Rule*.

Penelitian ini bertujuan menemukan pola assosiasi pembelian konsumen. Teknik pemodelan yang digunakan dalam penelitian ini adalah metode *Association Rules* sedangkan algoritma yang digunakan dalam penelitian ini adalah algoritma *FP-Growth* Dengan menerapkan *Data Mining*, data transaksi akan diolah dengan aturan asosiasi atau metode *Association Rules* untuk menemukan pola-pola pembelian produk yang sering dibeli oleh konsumen.

2. Build Model (Membangun Model)

Algoritma yang digunakan dalam penelitian ini adalah algoritma *FP-Growth*. Algoritma ini akan mencari himpunan menu yang sering dibeli dengan melihat *FP-Tree* yang sudah dibuat terlebih dahulu sehingga proses pencarian himpunan produk dapat lebih cepat dilakukan daripada algoritma apriori. Berikut ini adalah langkah-langkah proses pengerjaan algoritma *FP-Growth* :

a. Menghitung Frekuensi Kemunculan Tiap *Item*

Dengan menggunakan tabel 3.3, masing-masing *item* dihitung frekuensi kemunculan berdasarkan *item*nya. Hasil dari penghitungan kemunculan *item* dapat dilihat pada tabel 3.4 dibawah ini.

Tabel 3. 6 Hasi Perhitungan Kemunculan Produk

Nama Produk	Support Count
Wardah Facial Scrub 60 ml	16
PONDS CS ACNE C 40	13
PONDS CS ACNE C 40	13
CITRA HAZ SPOTLES WHITE GLOW UV 40 G	13
MAKARIZO SALON DAILY HAIR MASK 500gr NEW	12
La Tulip Milk Cleanser 120 ml New	10
PONDS FAIR LOVELY CREAM 25 G	10
KAPAS SARIAYU 35	9
PONDS CS ACNE CLEAR WHITE FF50 G	9
Pensil Alis Revlon	8
Pixy Eye Liner Black	8
PANTENE SHP ANTI DANRUFF 170 ML ATLAS	7
Wardah Lig Day Cream Step 1 30 gr	6
Wardah LIG NIGHT CREAM 20 step 2	6
WARDAH FACIAL SERUM NORMAL DRY SKIN	5
SUNSILK SHP SOFT 170 ML	5
WARDAH LIGHT BB CREAM LIGHT 15ML	5
REXONA ADVANCED WHITENING LOT 10G	4
ELIPS Hair Vit Repair	4
Wardah Excl. Liq Found. 02 Sheer Pink 20 ml	3
MINYAK KAYU PUTIH 120	3
HOT IN CREAM A.THERAPY 60 G	3
Eye LINER DAVIS WARNA	2
Just Miss Serut	1
Purbasari lips Daily Series 35	1

Awal mula ditentukan *minimum support* dan juga *confidence* dari keinginan pengguna untuk melihat batasan terendah munculnya menu. Penentuan *minimum support* sangat berpengaruh dengan hasil yang akan didapatkan. Apabila nilai *minimum*

support terlalu besar maka hasil yang didapatkan terlalu sedikit bahkan tidak ada, sedangkan apabila nilai *minimum support* nya terlalu kecil, hasil yang didapatkan terlalu banyak. Untuk kasus ini dimisalkan ditentukan *minimum support* : 4 dan *confidence* : 60%.. Hal ini dilakukan dengan memproses data yang ada pada tabel 3.6 yang dapat dilihat pada tabel 3.7.

Tabel 3. 7 Hasil Perhitungan *Support Item*

Kode	SUPPORT COUNT	Nilai Support
WFS60	16	59.25925926
WLCF60	14	51.85185185
PCSAC40	13	48.14814815
CHPWGUV40	13	48.14814815
MSDHM500	12	44.44444444
LTMC120	10	37.03703704
PFLC25	10	37.03703704
KS35	9	33.33333333
PCSACWFF50	9	33.33333333
PAR	8	29.62962963
PELB	8	29.62962963
PSHPAD170	7	25.92592593
WLDCS30S1	6	22.22222222
WLNC20S2	6	22.22222222
WFSNDS	5	18.51851852
SSHP170	5	18.51851852
WLBBCL15	5	18.51851852
RAWL10	4	14.81481481
EHVITR	4	14.81481481
WELFSP20	3	11.11111111
MKP120	3	11.11111111
HICAT60	3	11.11111111
ELDW	2	7.407407407
JMS	1	3.703703704
PLDS35	1	3.703703704

Langkah ke dua yaitu mengeleminasi menu yang tidak memenuhi *minimum support* 15%. Hasil Pengeleminasian menu dapat dilihat pada tabel 3.8.

Tabel 3. 8 Hasil Eleminasi *Minimum Support*

Kode	SUPPORT COUNT
WFS60	16
WLCF60	14
PCSAC40	13
CHPWGUV40	13
MSDHM500	12
LTMC120	10
PFLC25	10
KS35	9
PCSACWFF50	9
PAR	8
PELB	8
PSHPAD170	7
WLDCS30S1	6
WLNC20S2	6
WFSNDS	5
SSHP170	5
WLBBCL15	5
RAWL10	4
EHVITR	4

Langkah ketiga yaitu mengurutkan kembali *frequent* berdasarkan banyaknya kemunculan menu yang disebut dengan *Frequency of occurrence* yang dapat dilihat pada tabel 3.9.

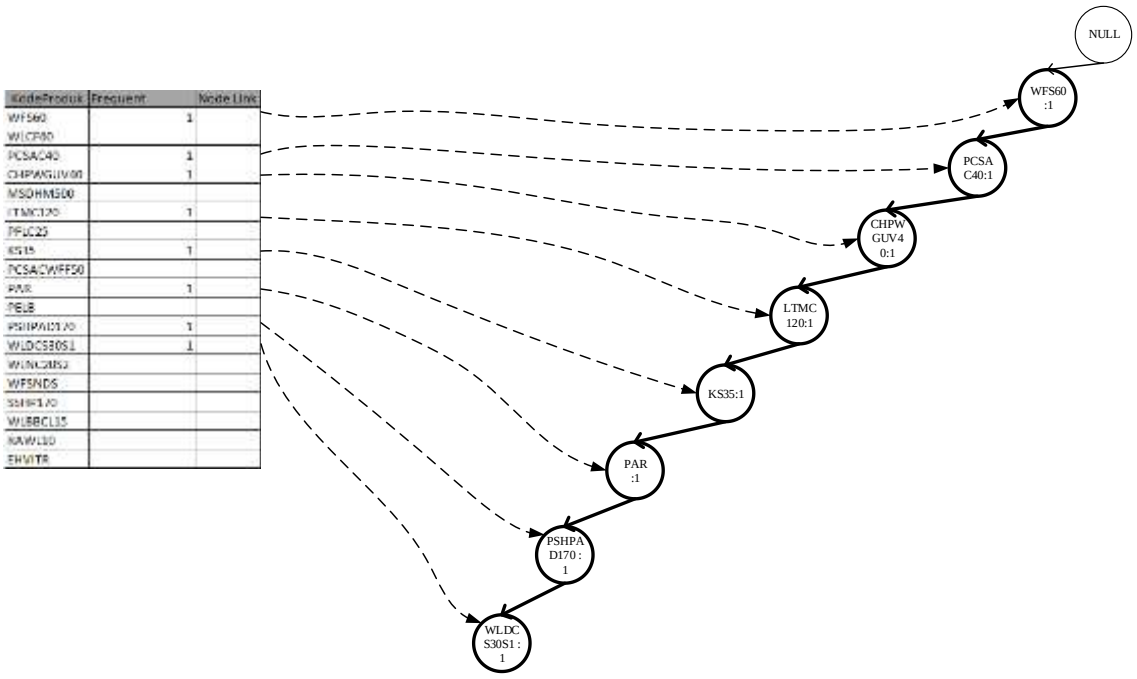
Tabel 3. 9 Priority

Kode	SUPPORT COUNT	Priority
WFS60	16	1
WLCF60	14	2
PCSAC40	13	3
CHPWGUV40	13	4
MSDHM500	12	5
LTMC120	10	6

PFLC25	10	7
KS35	9	8
PCSACWFF50	9	9
PAR	8	10
PELB	8	11
PSHPAD170	7	12
WLDCS30S1	6	13
WLNC20S2	6	14
WFSNDS	5	15
SSHP170	5	16
WLBBCL15	5	17
RAWL10	4	18
EHVITR	4	19

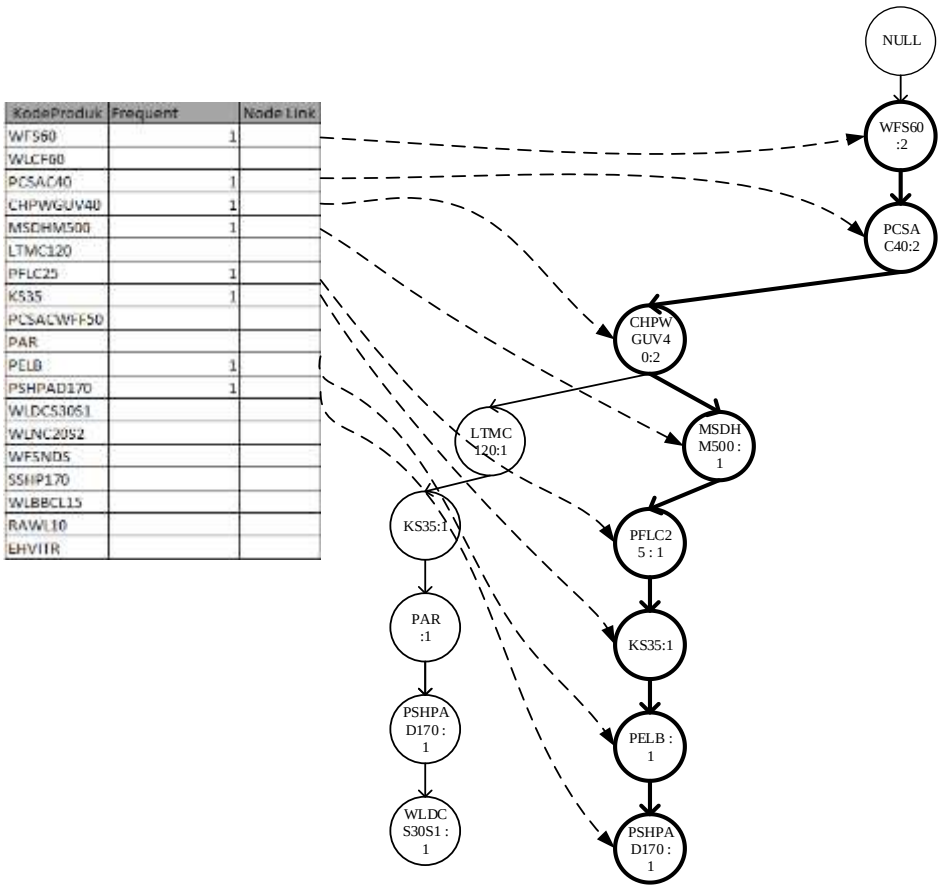
Langkah keempat adalah membuat *fp-tree*. Berikut proses pembentukan *FP-Tree* dibawah ini :

1) Transaksi 16010001 : WFS60, PCSAC40, CHPWGUV40, LTMC120, KS35, PAR, PSHPAD170, WLDCS30S1
2) Lakukan pembacaan pada data transaksi pertama yang akan diproses.
3) Lakukan pengecekan pada *root*yang diinisialisasikan dengan Null, jika root belum mempunyai anak (*child)*, maka data yang dibaca tadi menjadi anak pertama dari *root*. Dan *support count* dari masing masing *node* bertambah 1, kecuali *root*.



Gambar 3. 2 Tree Transaksi 16010001

1) Transaksi 16010002 : WFS60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25, KS35, PELB, PSHPAD170
    1) Lakukan pembacaan pada data transaksi kedua yang akan diproses.
    2) Jika data yang keluar berikutnya sama dengan data yang pertama, maka itu bukan anak kedua dari *root*, tetapi tetap anak dari *root* yang sebelumnya keluar, yang berubah adalah *support count*-nya menjadi bertambah 1.

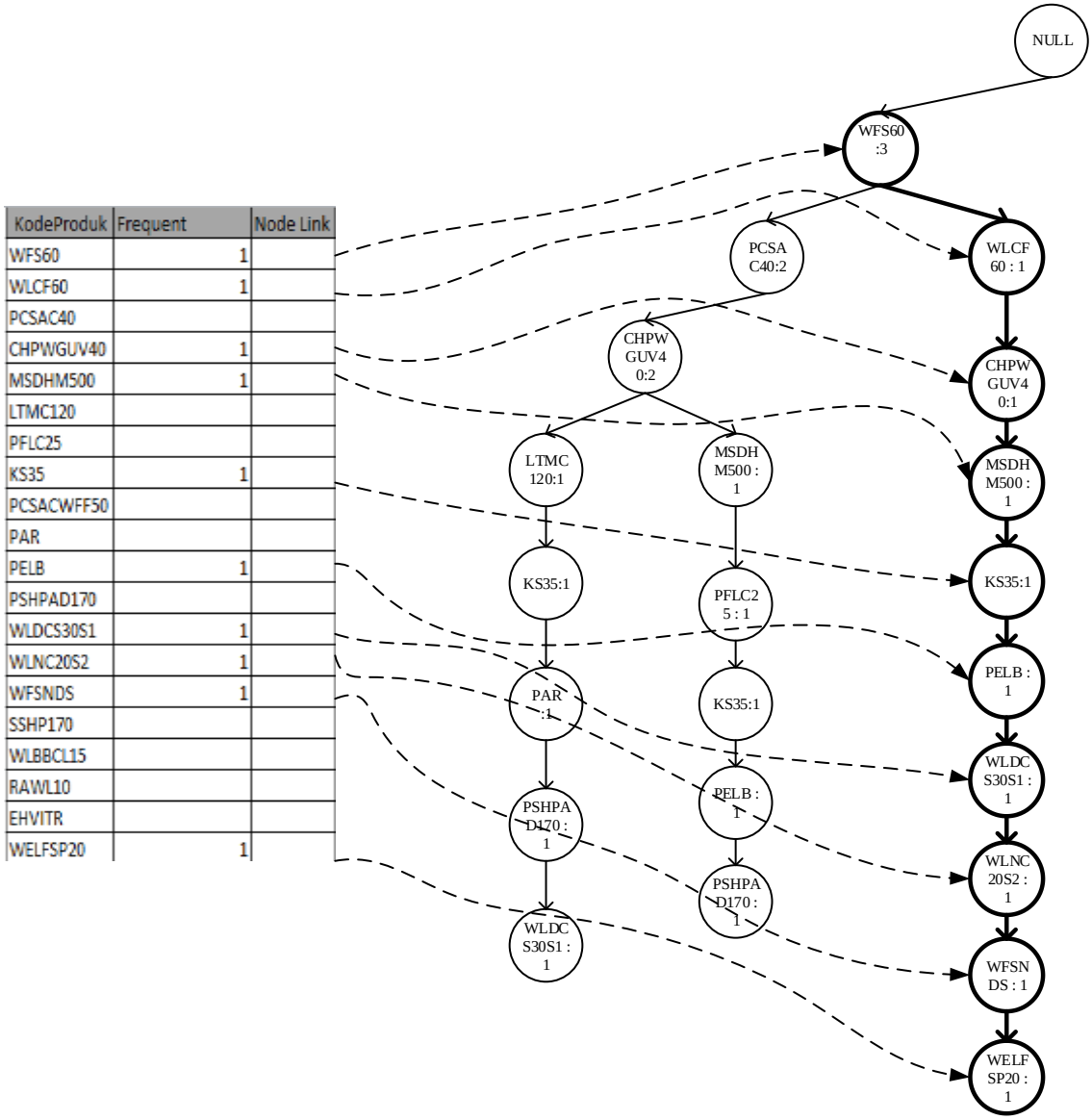


Gambar 3. 3 Tree Transaksi 16010002

2) Transaksi 16010003 : WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35, PELB, WLDCS30S1, WLNC20S2, WFSNDS, WELFSP20

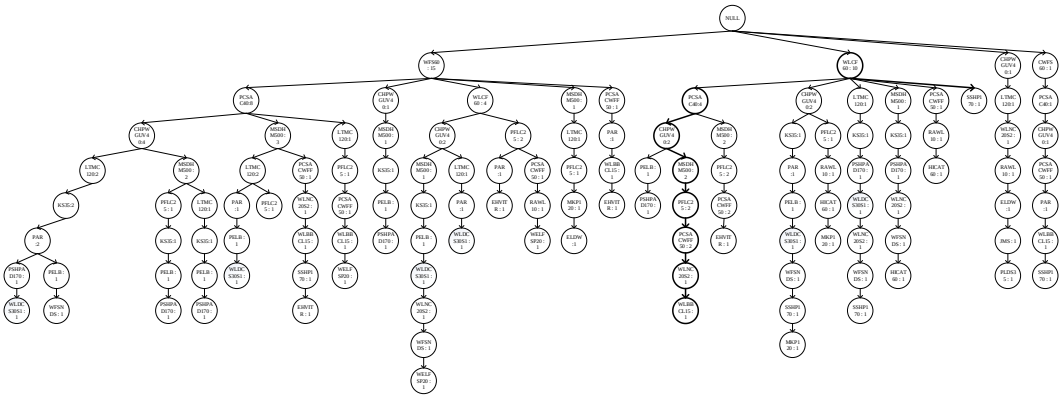
   1) Lakukan pembacaan pada data transaksi ketiga yang akan diproses.
   2) Jika data yang keluar sama setelah pembacaan, namun berbeda *lintasan*maka *nod*e pada *laintasan* yang lain tidak bertambah *support* count-nya, hanya data yang dibaca sama saja yang bertambah *support count*-nya.

KodeProduk	Frequent	Node Link
WFS60	1	
WLCF60	1	
PCSAC40		
CHPWGUV40	1	
MSDHM500	1	
LTMC120		
PFLC25		
KS35	1	
PCSACWFF50		
PAR		
PELB	1	
PSHPAD170		
WLDCS30S1	1	
WLNC20S2	1	
WFSNDS	1	
SSHP170		
WLBBCL15		
RAWL10		
EHVITR		
WELFSP20	1	



Gambar 3. 4 Tree Transaksi 16010003

3) Setelah pembacaan sampai ke transaksi 16010030 : WLCF60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25, PCSACWFF50, WLNC20S2, WLBBCL15 akan membentuk FP-tree di Gambar 3.5



Gambar 3. 5 Tree Transaksi 16010030

Setelah pembuatan *fp-tree* dilakukan langkah selanjutnya adalahTahap Pembangkitan *Conditional Pattern Base,* Tahap Pembangkitan *Conditional FP-Tree* dan pencarian Frequent Pattern. Adapun langkahnya yaitu :

a. Tahap Pembangkitan *Conditional Pattern Base*

   Mencari *support count* terkecil dari setiap *Item* sampai dengan terbesar berdasarkan hasil pembentukan *fp-tree* sebelumnya . Berikut ini adalah *support count* terkecil sampai dengan terbesar dapat dilihat pada table 3.10

Tabel 3. 10 Conditonal Pattern Base

Item	***Conditonal Pattern Base***
EHVITR	• WFS60, PCSAC40, MSDHM500, PCSACWFF50, WLNC20S2, WLBBCL15, SSHP170 : 1 • WFS60, WLCF60, PFLC25, PAR : 1 • WFS60, PCSAWFF50. PAR, WLBBCL : 1 • WLCF60, PCSAC40, MSDHM500, PFLC25, PCSAWFF50: 1
RAWL10	• WFS60, WLCF60, PFLC25, PCSACWFF50 : 1 • WLCF60, CHPWGUV40, PFLC25 : 1 • WLCF60, PCSACWFF50 : 1 • CHPWGUV40, LTMC120, WLNC20S2 : 1
WLBBCL15	• WFS60, PCSAC40, CHPWGUV40, PCSACWFF50, PAR : 1 • WFS60, PCSAC40, MSDHM500, PCSACWFF50, WLNC20S2 : 1 • WFS60, PCSAC40, LTMC120, PFLC25, PCSACWFF50 : 1 • WFS60, PCSACWFF50, PAR : 1 • WLCF60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25, PCSACWFF50, WLNC20S2 : 1
WFSNDS	• WFS60, PCSAC40, CHPWGUV40, LTMC120, KS35, PAR, PELB : 1

	• WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35, PELB, WLDCS30S1, WLNC20S2 : 1 • WLCF60, CHPWGUV40, KS35, PAR, PELB, WLDCS30S1 : 1 • WLCF60, LTMC120, KS35, PSHPAD170, WLDCS30S1, WLNC20S2 : 1 • WLCF60, MSDH500, KS35, PSHPAD170, WLNC20S2 : 1
WLNC20S2	• WFS60, PCSA40, MSDHM500, PCSACWFF50:1 • WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35, PELB, WLDCS30S1:1 • WLCF60, PCSA40, CHPWGUV40, MSDHM500, PFLC2, PCSACWFF50:1 • WLCF60, LTMC120, KS35, PSHPAD170, WLDCS30S1: 1 • WLCF60, MSDH500, KS35, PSHPAD170 : 1 • CHPWGUV40, LTMC120 : 1
WLDCS30S1	• WFS60, PCSAC40, CHPWGUV40, LTMC120, KS35, PAR, PSHPAD170 : 1 • WFS60, PCSAC40, MSDHM500, LTMC120, PAR,PELB : 1 • WFS60, WLCF60, CHPWGUV40,MSDHM500,KS35, PELB : 1 • WFS60, WLCF60, CHPWGUV40,LTMC120, PAR : 1 • WLCF60, CHPWGUV40, KS35, PAR, PELB : 1 • WLCF60, LTMC120, KS35, PSHPAD170 : 1
PSHPAD170	• WFS60, PCSAC40,CHPWGUV40,LTMC120, KS35, PAR : 1 • WFS60, PCSAC40,CHPWGUV40,MSDHM500, PFLC2, KS35, PELB : 1

	• WFS60, PCSAC40,CHPWGUV40,MSDHM500, LTMC120, KS35, PELB : 1 • WFS60, CHPWGUV40, MSDHM500, KS35, PELB : 1 • WLCF60, PCSAC40,CHPWGUV40, PELB : 1 • WLCF60, LTMC120, KS35 : 1 • WLCF60, MSDHM500, KS35 : 1
PELB	• WFS60, PCSAC40,CHPWGUV40,LTMC120, KS35, PAR : 1 • WFS60, PCSAC40,CHPWGUV40,MSDHM500, PFLC2, KS35 ; 1 • WFS60, PCSAC40,CHPWGUV40,MSDHM500, LTMC120, KS35 ; 1 • WFS60, PCSAC40,MSDHM500, LTMC120, PAR ; 1 • WFS60, CHPWGUV40, MSDHM500, KS35, : 1 • WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35 : 1 • WLCF60, PCSAC40,CHPWGUV40 : 1 • WLCF60, CHPWGUV40, KS35, PAR : 1
PCSACWFF50	• WFS60, PCSAC40, MSDHM500 : 1 • WFS60, PCSAC40,LTMC120, PFLC25 : 1 • WFS60, PCSAC40, CHPWGUV40 : 1 • WFS60, WLCF60, PFLC25 : 1 • WFS60 : 1 • WLCF60, PCSAC40, CHPWGUV40, MSDHM500, PLFC25 : 1 • WLCF60, PCSAC40, MSDHM500, PLFC25 : 2 • WLCF60 : 1
KS35	• WFS60, PCSAC40, CHPWGUV40, LTMC120 : 2 • WFS60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25 : 1 • WFS60, PCSAC40, CHPWGUV40, MSDHM500, LTMC120 : 1

	• WFS60, CHPWGUV40, MSDHM500 : 1 • WFS60, WLCF60, CHPWGUV40, MSDHM500 : 1 • WLCF60, CHPWGUV40 : 1 • WLCF60, LTMC120 : 1 • WLCF60, MSDHM500 : 1
PFLC25	• WFS60, PCSAC40, CHPWGUV40, MSDHM500 : 1 • WFS60, PCSAC40, MSDHM500, LTMC120 : 1 • WFS60, PCSAC40, LTMC120 : 1 • WFS60, WLCF60 : 2 • WFS60, MSDHM500, LTMC120 : 1 • WLCF60, PCSAC40, CHPWGUV40, MSDHM500 : 1 • WLCF60, PCSAC40, MSDHM500 : 2 • WLCF60, CHPWGUV40 : 1
LTMC120	• WFS60, PCSAC40, CHPWGUV40 : 2 • WFS60, PCSAC40, CHPWGUV40, MSDHM500 : 1 • WFS60, PCSAC40, MSDHM500 : 2 • WFS60, PCSAC40 : 1 • WFS60, WLCF60, CHPWGUV40 : 1 • WFS60, MSDHM500 : 1 • WLCF60: 1 • CHPWGUV40 : 1
MSDHM500	• WFS60, PCSAC40, CHPWGUV40 : 2 • WFS60, PCSAC40 : 3 • WFS60, CHPWGUV40 : 1 • WFS60, WLCF60, CHPWGUV40 : 1 • WFS60 : 1 • WLCF60, PCSAC40, CHPWGUV40 : 1 • WLCF60, PCSAC40 : 2 • WLCF60 : 1
CHPWGUV40	• WFS60, PCSAC40 : 5 • WFS60 : 1

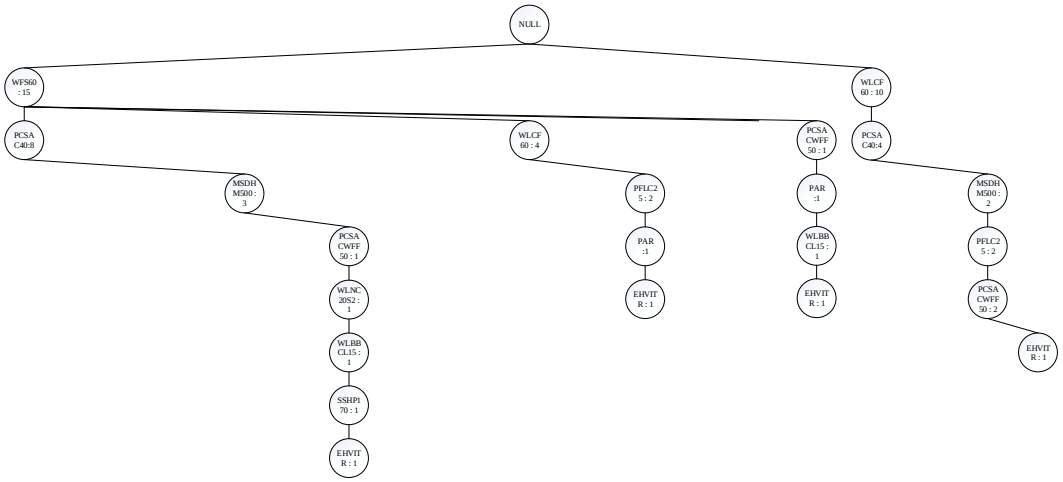
	• WFS60, WLCF60 : 2 • WLCF60, PCSAC40 : 2 • WLCF60 : 2 • CHPWGUV40 : 1
PCSAC40	• WFS60 : 9 • WLCF60 : 4
WLCF60	• WFS60: 4

b. Tahap Pembangkitan *Conditional FP-Tree*

Pada tahap ini setiap *Item-Item*akan dibuat *conditional fp-tree*, dan apabila ada *Item* yang memenuhi *minimum support* atau sama dengan 6, maka *Item* tersebut akan dibangkitkan lagi untuk mencari *frequent pattern*. Berikut adalah *conditional fp-tree* dari tiap *Item.*

1. *Conditional FP-TreeItem EHVITR* :

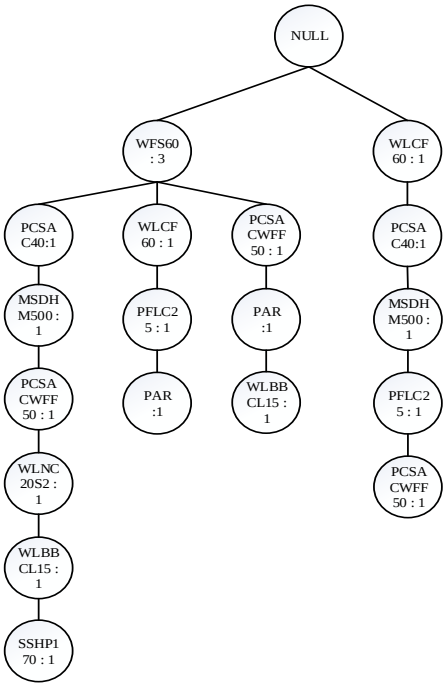
Tahap pembentukan *Conditional* FP-Tree untuk *Item* EHVITR dengan *Conditional pattern base* {WFS60, PCSAC40, MSDHM500, PCSACWFF50, WLNC20S2, WLBBCL15, SSHP170 : 1}, {WFS60, WLCF60, PFLC25, PAR : 1}, {WFS60, PCSAWFF50. PAR, WLBBCL : 1}, {WLCF60, PCSAC40, MSDHM500, PFLC25, PCSAWFF50: 1}



Gambar 3. 6 *Conditional FP-TreeItem EHVITR*

Karena *support count Item EHVITR* memenuhi *minimum support* maka *Item EHVITR* akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

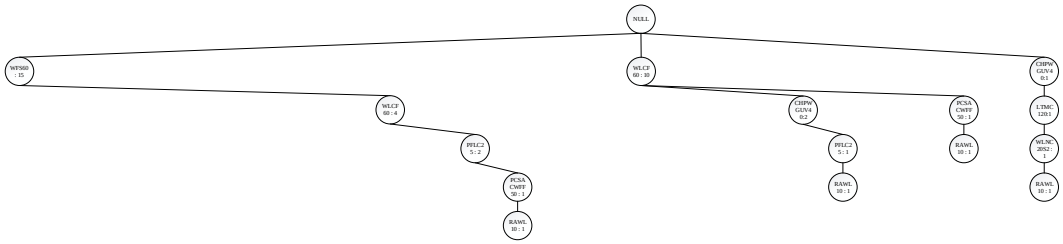
1. *Item* { *EHVITR* } akan dihilangkan dan *support count* yang satu lintasan dengan { *EHVITR* } akan sama dengan *support count* { *EHVITR* } sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 7 *Conditional FP-TreeItem EHVITR* setelah dibangkitkan

2. *Conditional FP-TreeItem* RAWL10 *:*

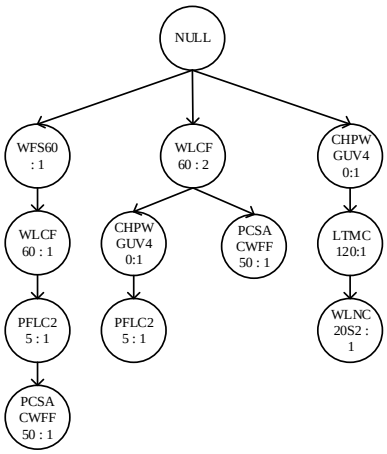
Tahap pembentukan *Conditional* FP-Tree untuk *Item RAWL10* dengan *Conditional pattern base* {WFS60, WLCF60, PFLC25, PCSACWFF50 : 1}, {WLCF60, CHPWGUV40, PFLC25 : 1}, {WLCF60, PCSACWFF50 : 1}, {CHPWGUV40, LTMC120, WLNC20S2 : 1}



Gambar 3. 8 *Conditional FP-TreeItem* RAWL10

Karena *support count Item* RAWL10 memenuhi *minimum support* maka *Item RAWL10* akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

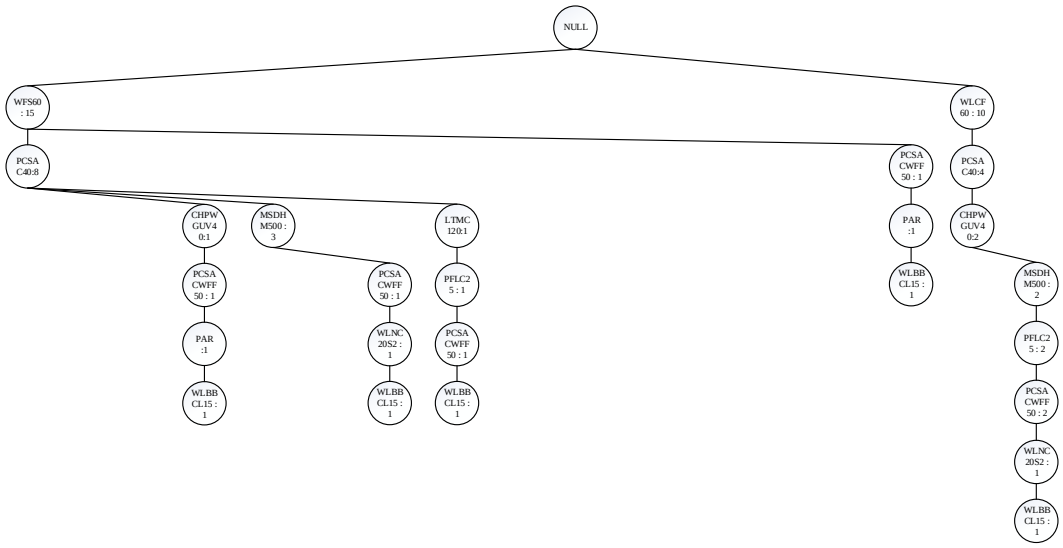
1. *Item* { RAWL10 } akan dihilangkan dan *support count* yang satu lintasan dengan { RAWL10 } akan sama dengan *support count* { RAWL10 } sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 9 *Conditional FP-TreeItem* RAWL10 setelah dibangkitkan

3. *Conditional FP-TreeItem* WLBBCL15 :

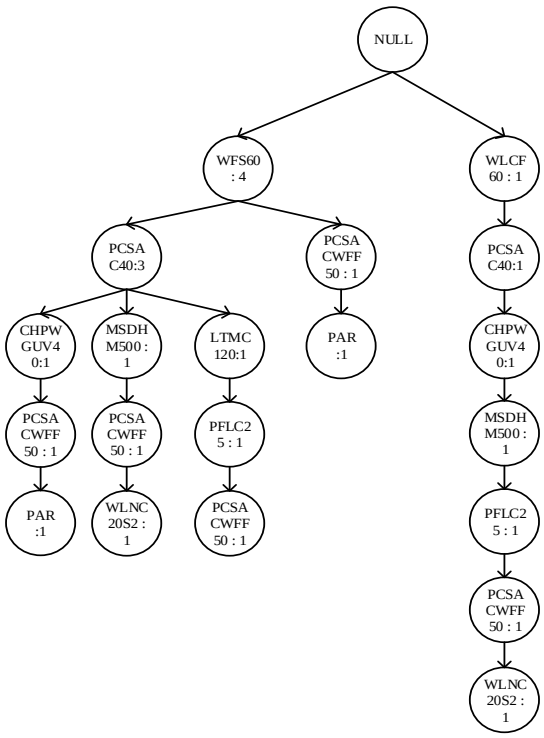
Tahap pembentukan *Conditional* FP-Tree untuk *Item* WLBBCL15 dengan *Conditional pattern base* {WFS60, PCSAC40, CHPWGUV40, PCSACWFF50, PAR : 1}, {WFS60, PCSAC40, MSDHM500, PCSACWFF50, WLNC20S2 : 1}, {WFS60, PCSAC40, LTMC120, PFLC25, PCSACWFF50 : 1}, {WFS60, PCSACWFF50, PAR : 1}, {WLCF60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25, PCSACWFF50, WLNC20S2 : 1}



Gambar 3. 10 *Conditional FP-TreeItem* WLBBCL15

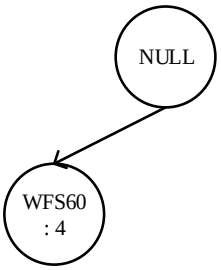
Karena *support count Item* WLBBCL15 memenuhi *minimum support* maka *Item* WLBBCL15 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1. *Item* { WLBBCL15 } akan dihilangkan dan *support count* yang satu lintasan dengan { WLBBCL15 } akan sama dengan *support count* { WLBBCL15 } sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

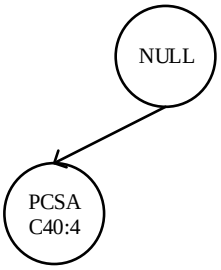


Gambar 3. 11 *Conditional FP-Tree Item* WLBBCL15 setelah dibangkitkan

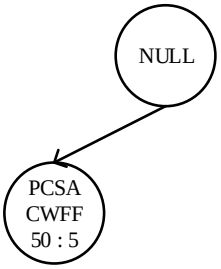
3. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { WFS60 : 4}, {PCSA40 : 4}, {PCSACWFF50 : 5} adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 12 *Conditional FP-Tree* WLBBC15, WFS60 : 4



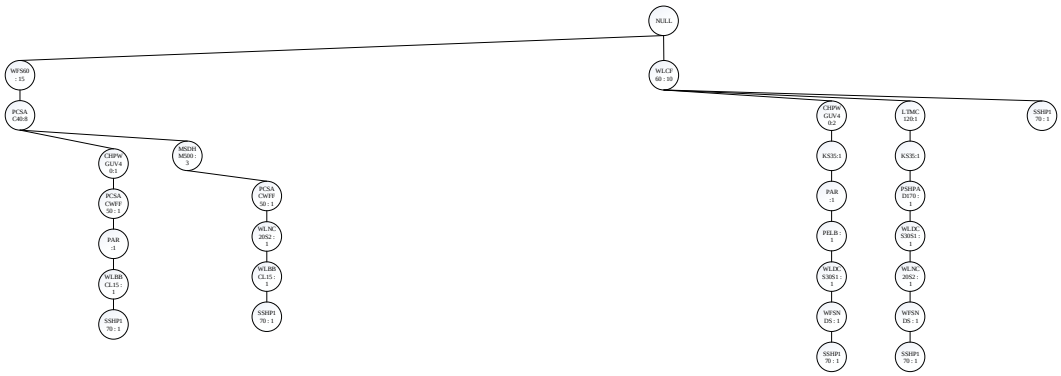
Gambar 3. 13 *Conditional FP-Tree* WLBBC15, PCSAC40 : 4



Gambar 3. 14 *Conditional FP-Tree* WLBBC15, PCSACWFF50 : 5

4. *Conditional FP-TreeItem SSHP1705* :

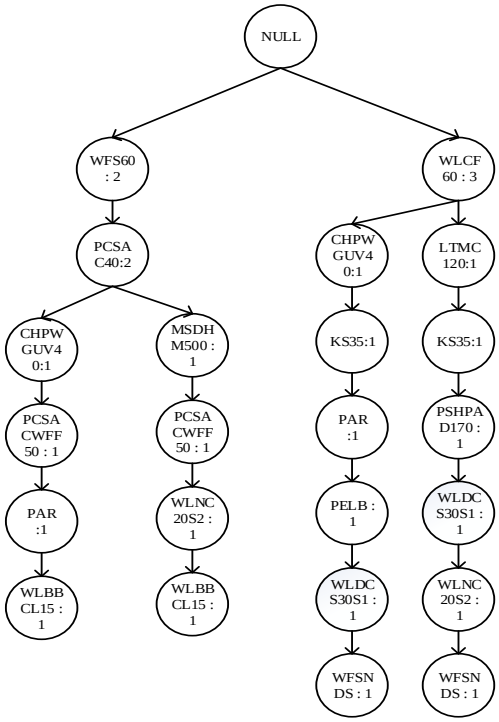
Tahap pembentukan *Conditional* FP-Tree untuk *Item* SSHP170 dengan *Conditional pattern base* {WFS60, PCSAC40, CHPWGUV40, PCSACWFF50, PAR, WLBBCL15 : 1}, {WFS60, PCSAC40, MSDHM500, PCSACWFF50, WLNC20S2, WLBBCL15: 1}, {WLCF60, CHPWGUV40, KS35, PAR, PELB, WLDCS30S1, WFSNDS : 1}, {WLCF60, LTMC120, KS35, PSHPAD170, WLDCS30S1, WLNC20S2, WFSNDS : 1},{WLCF60 : 1}



Gambar 3. 15 *Conditional FP-TreeItem* SSHP1705

Karena *support count Item SSHP1705* memenuhi *minimum support* maka *Item SSHP1705* akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1. *Item* { *SSHP1705* } akan dihilangkan dan *support count* yang satu lintasan dengan { *SSHP1705* } akan sama dengan *support count* { *SSHP1705* } sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

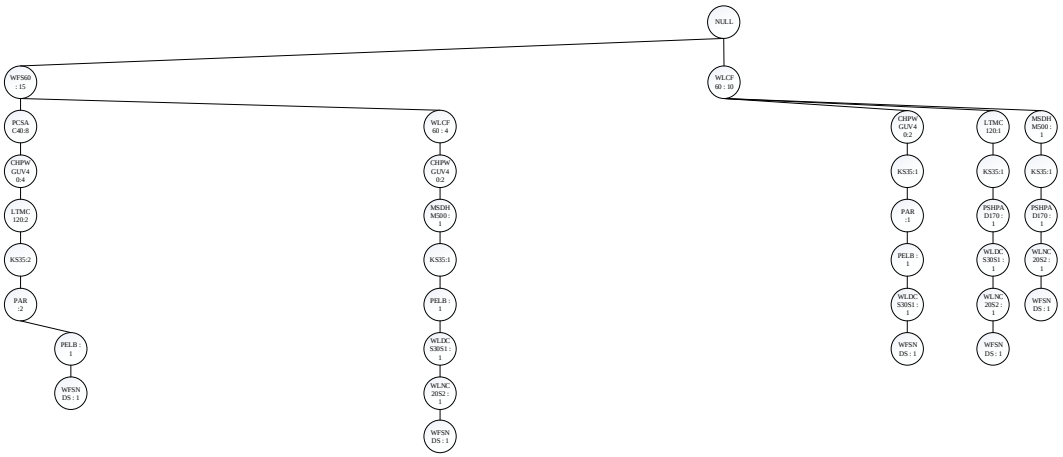


Gambar 3. 16 *Conditional FP-TreeItem SSHP1705* setelah dibangkitkan

5. *Conditional FP-TreeItem* WFSNDS:

Tahap pembentukan *Conditional* FP-Tree untuk *Item* WFSNDS dengan *Conditional pattern base* {WFS60, PCSAC40, CHPWGUV40, LTMC120, KS35, PAR, PELB : 1}, {WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35, PELB, WLDCS30S1, WLNC20S2 : 1}, {WLCF60, CHPWGUV40,

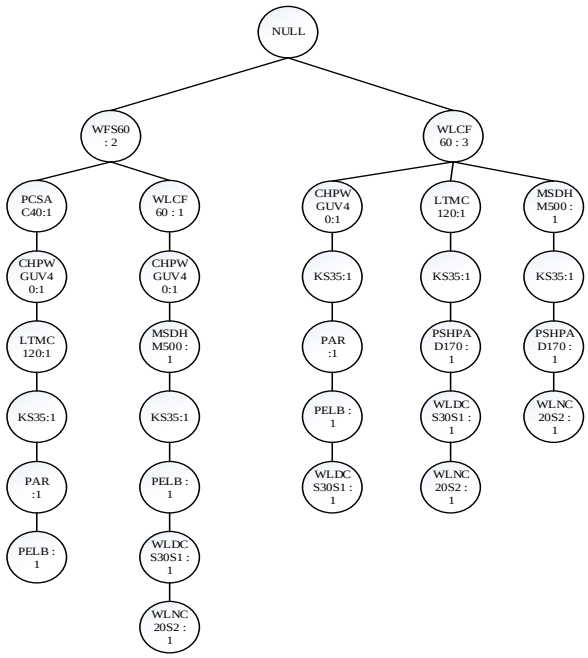
KS35, PAR, PELB, WLDCS30S1 : 1}, {WLCF60, LTMC120, KS35, PSHPAD170, WLDCS30S1, WLNC20S2 : 1}, {WLCF60, MSDH500, KS35, PSHPAD170, WLNC20S2 : 1}



Gambar 3. 17 *Conditional FP-TreeItem* WFSNDS

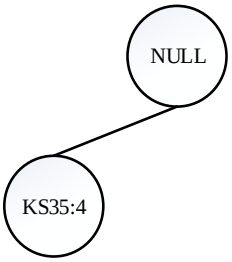
Karena *support count Item* WFSNDS memenuhi *minimum support* maka *Item* WFSNDS akan dibangkitkan dengan *conditional fp-tree,* berikut langkah langkahnya :

1. *Item* { WFSNDS } akan dihilangkan dan *support count* yang satu lintasan dengan { WFSNDS } akan sama dengan *support count* { WFSNDS } sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

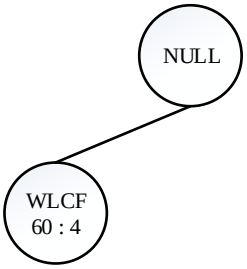


Gambar 3. 18 *Conditional FP-TreeItem* WFSNDS setelah dibangkitkan

3. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { KS35 : 4}, {WLCF60 : 4} adalah *Item* yang keluar yang dapat memenuhi minimum support



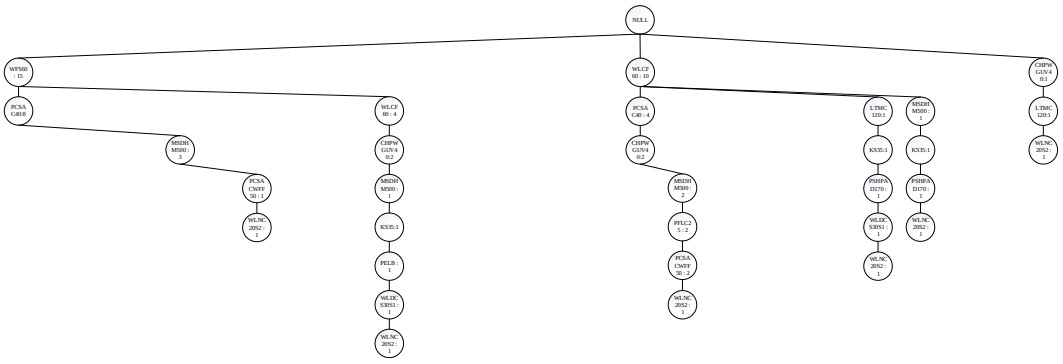
Gambar 3. 19 *Conditional FP-Tree* WFSNDS, KS35 : 4



Gambar 3. 20 *Conditional FP-Tree* WFSNDS, WLCF60 : 4

6. *Conditional FP-TreeItem* WLNC20S2 :

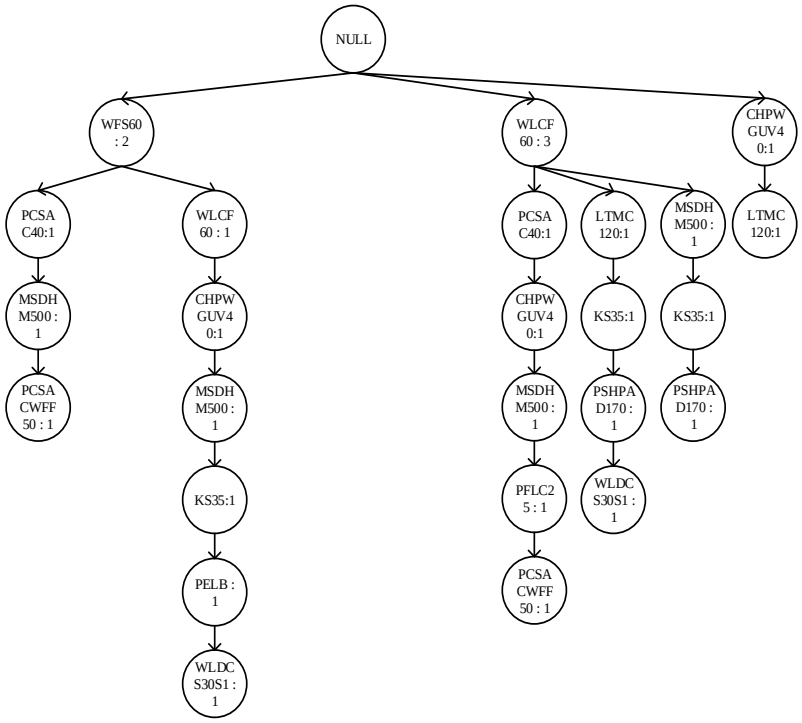
Tahap pembentukan *Conditional* FP-Tree untuk *Item* WLNC20S2 dengan *Conditional pattern base* {WFS60, PCSA40, MSDHM500, PCSACWFF50:1},{WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35, PELB, WLDCS30S1:1},{WLCF60, PCSA40, CHPWGUV40, MSDHM500, PFLC2, PCSACWFF50:1},{WLCF60, LTMC120, KS35, PSHPAD170, WLDCS30S1: 1},{WLCF60, MSDH500, KS35, PSHPAD170 : 1},{CHPWGUV40, LTMC120 : 1}



Gambar 3. 21 *Conditional FP-TreeItem WLNC20S2*

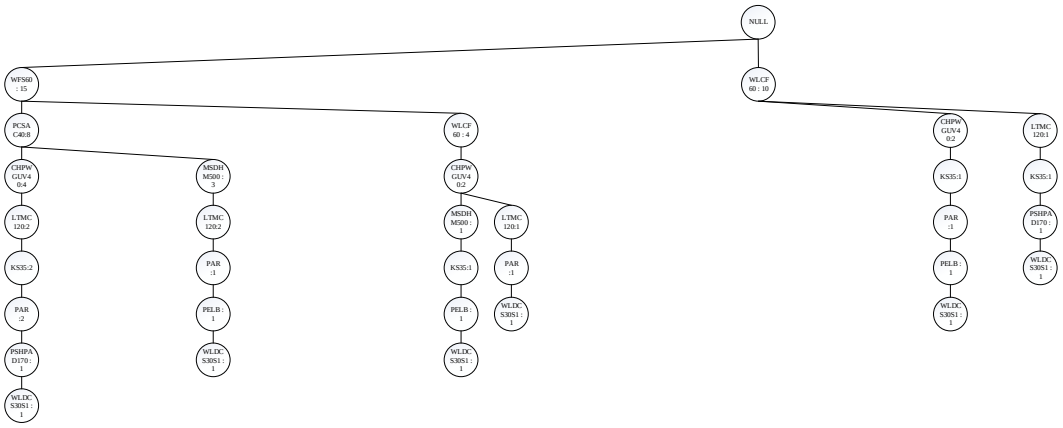
Karena *support count Item* WLNC20S2 memenuhi *minimum support* maka *Item* WLNC20S2 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1. *Item* {WLNC20S2} akan dihilangkan dan *support count* yang satu lintasan dengan {WLNC20S2} akan sama dengan *support count* {WLNC20S2} sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 22 *Conditional FP-TreeItem WLNC20S2 setelah dibangkitkan*

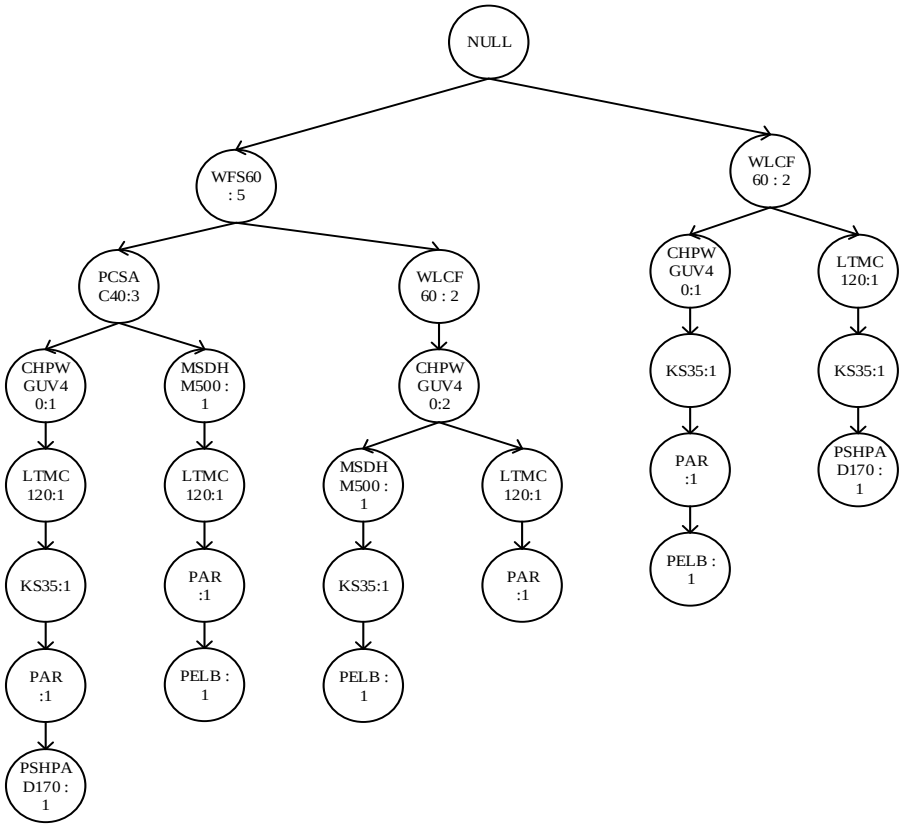
1. *Conditional FP-Tree*untuk*Item* WLDCS30S1 dengan *conditional pattern base* {WFS60, PCSAC40, CHPWGUV40, LTMC120, KS35, PAR, PSHPAD170 : 1},{WFS60, PCSAC40, MSDHM500, LTMC120, PAR,PELB : 1},{WFS60, WLCF60, CHPWGUV40,MSDHM500,KS35, PELB : 1},{ WFS60, WLCF60, CHPWGUV40,LTMC120, PAR : 1},{ WLCF60, CHPWGUV40, KS35, PAR, PELB : 1},{WLCF60, LTMC120, KS35, PSHPAD170 : 1}



Gambar 3. 23 *Conditional FP-TreeItem* WLDCS30S1

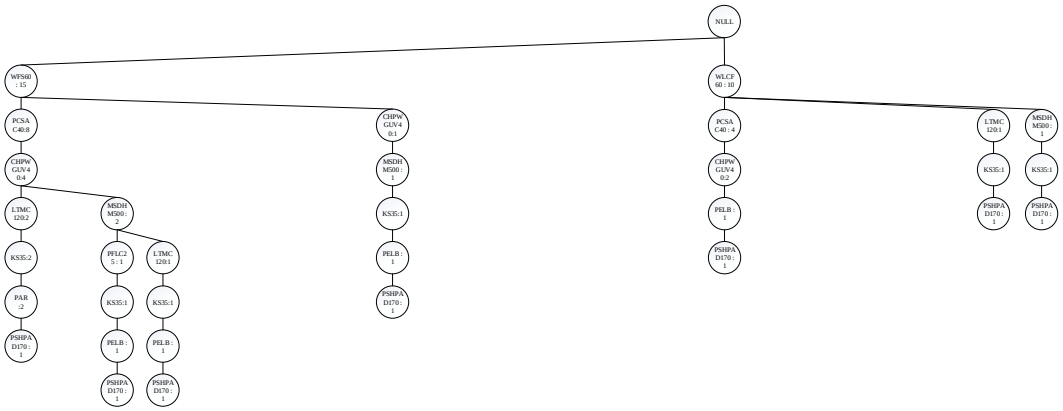
Karena *support count Item* WLDCS30S1 memenuhi *minimum support* maka *Item* WLDCS30S1 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1. *Item* { WLDCS30S1} akan dihilangkan dan *support count* yang satu lintasan dengan { WLDCS30S1} akan sama dengan *support count* { WLDCS30S1} sebelumnya.
2. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 24 *Conditional FP-TreeItem* WLDCS30S1 setelah dibangkitkan

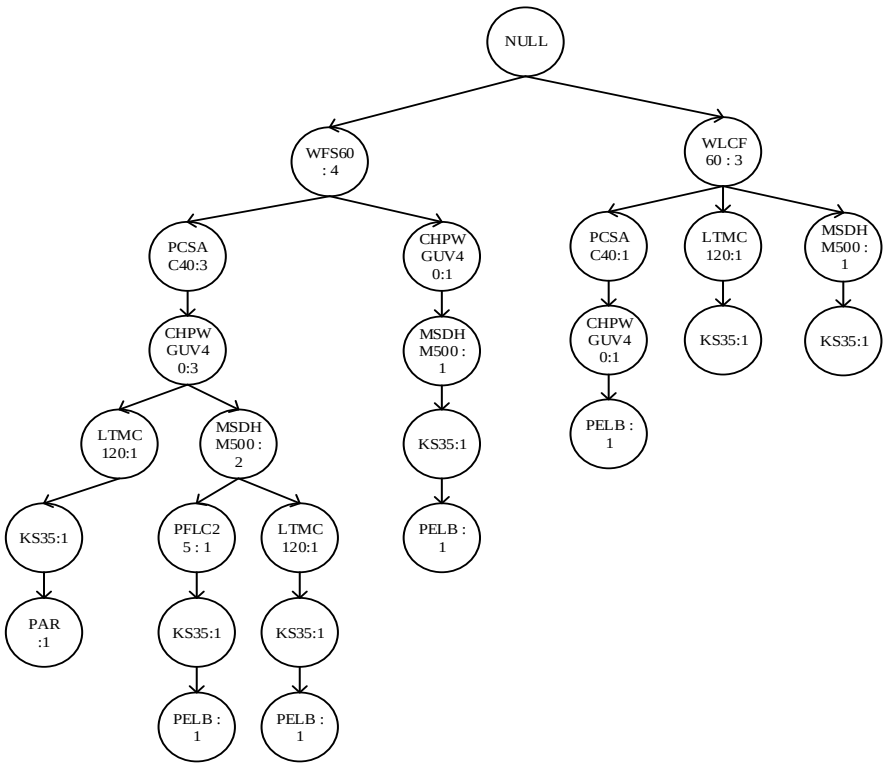
2. *Conditional FP-Tree*untuk*Item* PSHPAD170 dengan *conditional pattern base* {WFS60, PCSAC40,CHPWGUV40,LTMC120, KS35, PAR : 1},{WFS60, PCSAC40,CHPWGUV40,MSDHM500, PFLC2, KS35, PELB : 1},{WFS60, PCSAC40,CHPWGUV40,MSDHM500, LTMC120, KS35, PELB : 1},{WFS60, CHPWGUV40, MSDHM500, KS35, PELB : 1},{WLCF60, PCSAC40,CHPWGUV40, PELB : 1},{WLCF60, LTMC120, KS35 : 1},{WLCF60, MSDHM500, KS35 : 1}



Gambar 3. 25 *Conditional FP-TreeItem* PSHPAD170

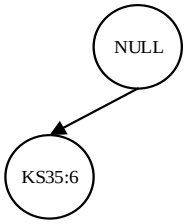
Karena *support count Item* PSHPAD170 memenuhi *minimum support* maka *Item* PSHPAD170 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

3. *Item* { PSHPAD170 } akan dihilangkan dan *support count* yang satu lintasan dengan { PSHPAD170 } akan sama dengan *support count* { PSHPAD170 } sebelumnya.
7. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 26 *Conditional FP-TreeItem* PSHPAD170 setelah dibangkitkan

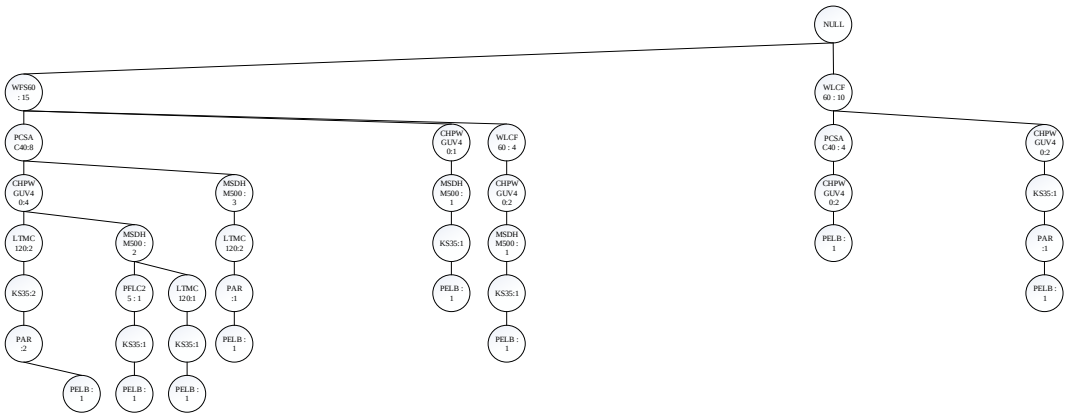
8. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { KS35:6} adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 27 *Conditional FP-Tree* PSHPAD170, KS35 :6

*1. Conditional FP-Tree*untuk*Item* PELB dengan *conditional pattern base* {WFS60, PCSAC40,CHPWGUV40,LTMC120, KS35, PAR : 1},{WFS60, PCSAC40,CHPWGUV40,MSDHM500, PFLC2, KS35 : 1},{.WFS60, PCSAC40,CHPWGUV40,MSDHM500, LTMC120, KS35 : 1},{ WFS60, PCSAC40,MSDHM500, LTMC120, PAR : 1},{.WFS60, CHPWGUV40,

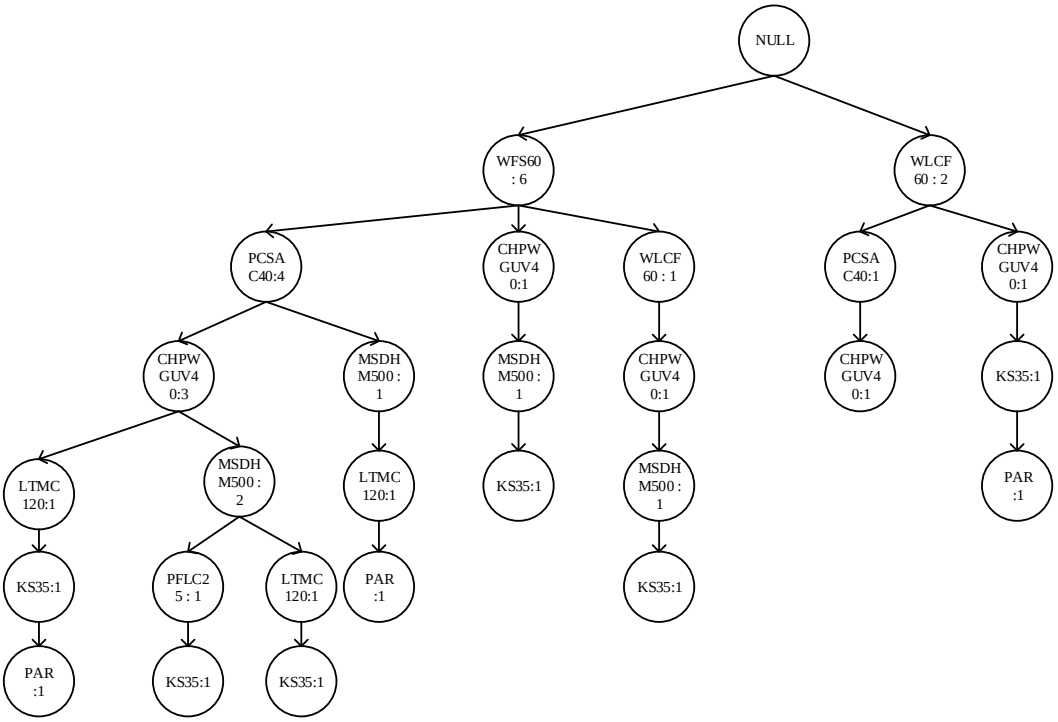
MSDHM500, KS35, : 1},{.WFS60, WLCF60, CHPWGUV40, MSDHM500, KS35 :1},{. WLCF60, PCSAC40,CHPWGUV40:1},{WLCF60, CHPWGUV40, KS35, PAR:1}



Gambar 3. 28 *Conditional FP-TreeItem* PELB

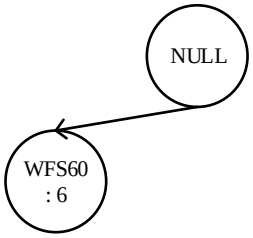
Karena *support count Item* PELB memenuhi *minimum support* maka *Item* PELB akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

2. *Item* { PELB } akan dihilangkan dan *support count* yang satu lintasan dengan { PELB } akan sama dengan *support count* { PELB } sebelumnya.
3. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

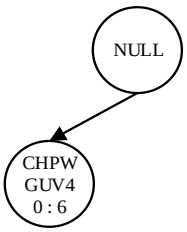


Gambar 3. 29 *Conditional FP-TreeItem* PELB setelah dibangkitkan

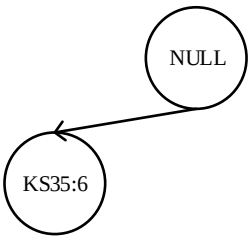
4. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { WFS60:6},{ CHPWGUV40 : 6}, {KS35 : 6} adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 30 *Conditional FP-Tree* PELB , WFS60:6

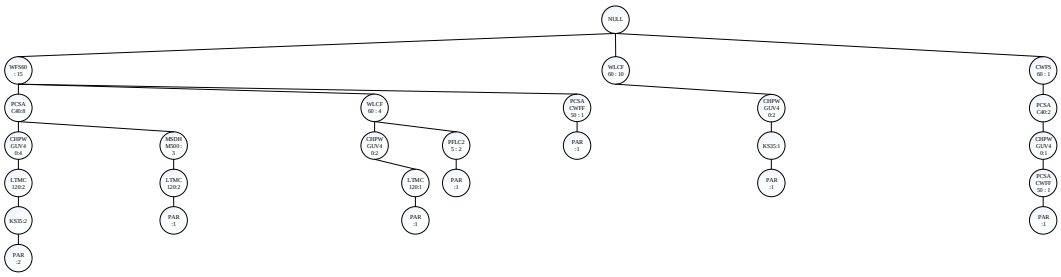


Gambar 3. 31 *Conditional FP-Tree* PELB , CHPWGUV40 : 6



Gambar 3. 32 *Conditional FP-Tree* PELB , KS35 : 6

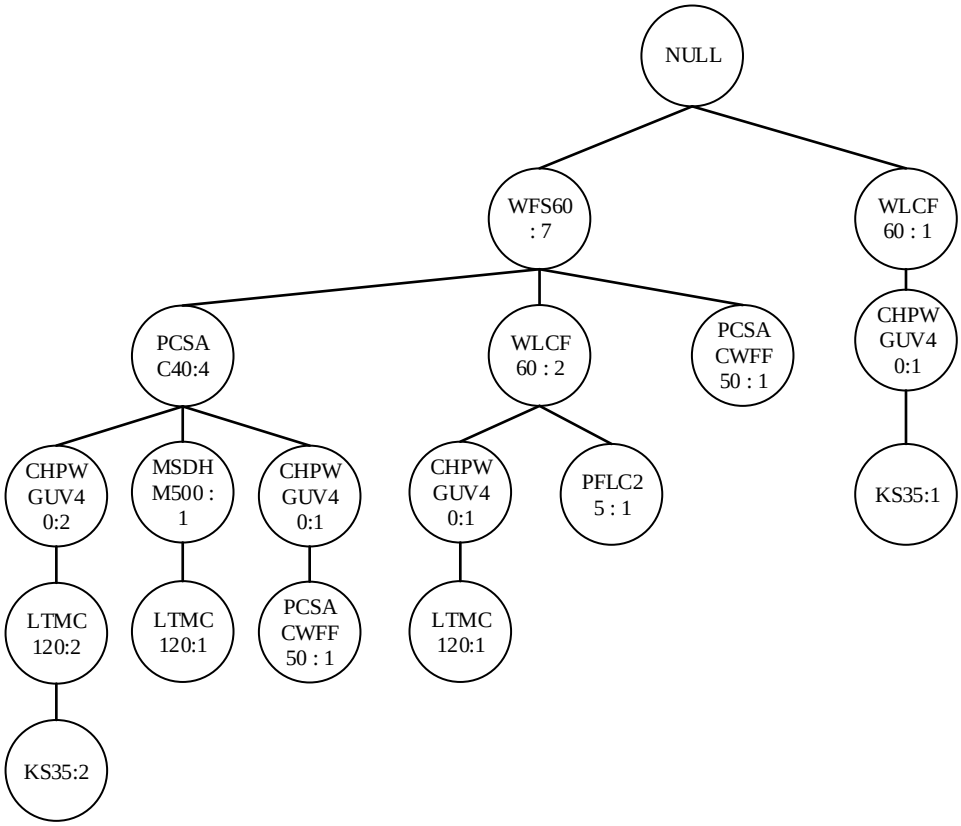
1. *Conditional FP-Tree*untuk*Item* PAR dengan *conditional pattern base* {WFS60, PCSAC40,CHPWGUV40,LTMC120, KS35 : 2}, { WFS60, PCSAC40,MSDHM500, LTMC120 : 1}, { WFS60, PCSAC40,CHPWGUV40, PCSACWFF50 : 1}, { WFS60, WLCF60, CHPWGUV40, LTMC120 : 1}, {WFS60, WLCF60, PFLC25 : 1},{ WFS60, PCSACWFF50 : 1},{ WLCF60, CHPWGUV40, KS35 : 1}



Gambar 3. 33 *Conditional FP-TreeItem* PAR

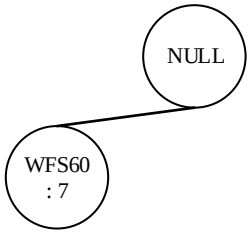
Karena *support count Item* PAR memenuhi *minimum support* maka *Item* PAR akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

2. *Item* { PAR } akan dihilangkan dan *support count* yang satu lintasan dengan { PAR } akan sama dengan *support count* { PAR } sebelumnya.
3. Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 34 *Conditional FP-TreeItem* PAR setelah dibangkitkan

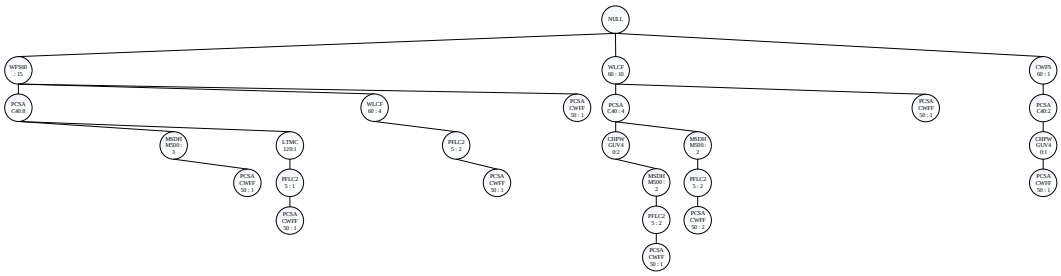
b. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { WFS60:7} adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 35 *Conditional FP-Tree* PAR , WFS60:7 Hasil *Frequent pattern*

6. *Conditional FP-Tree*untuk*Item* PCSACWFF50 dengan *conditional pattern base* {WFS60, PCSAC40, MSDHM500 : 1}, {WFS60, PCSAC40,LTMC120,

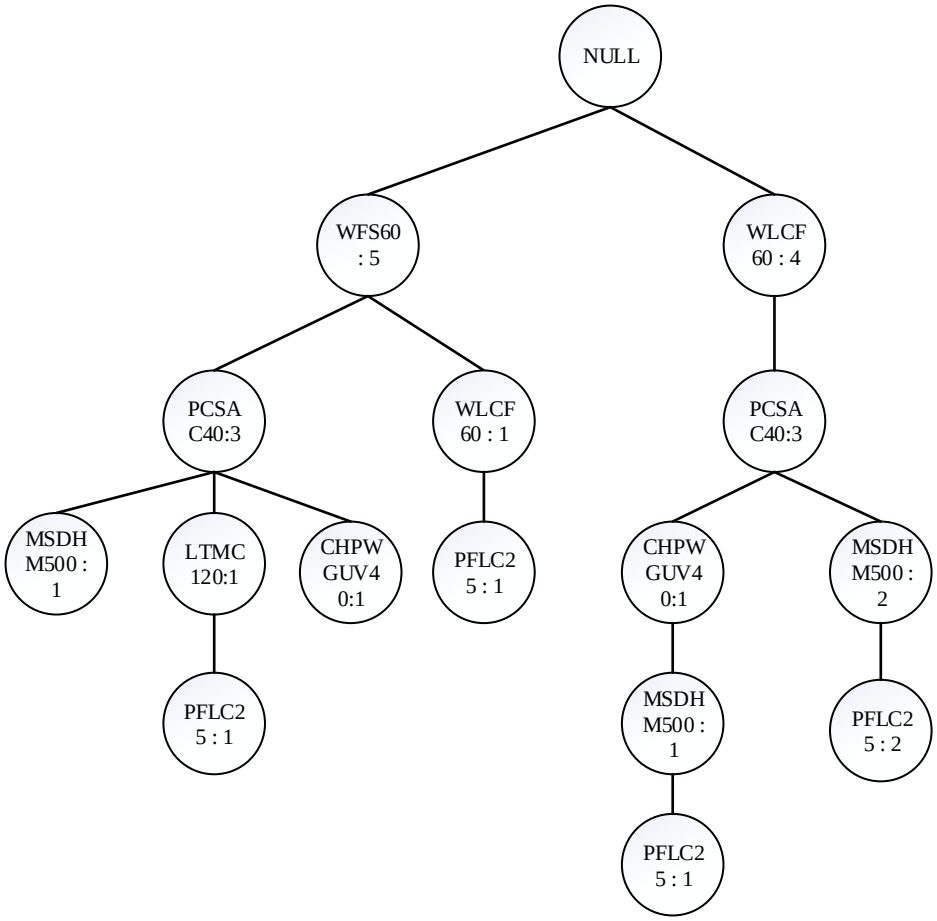
PFLC25 : 1},{ WFS60, PCSAC40, CHPWGUV40 : 1},{WFS60, WLCF60, PFLC25 : 1},{ WFS60 : 1},{WLCF60, PCSAC40, CHPWGUV40, MSDHM500, PLFC25 : 1},{WLCF60, PCSAC40, MSDHM500, PLFC25 : 2},{WLCF60 : 1}.



Gambar 3. 36 *Conditional FP-TreeItem* PCSACWFF50

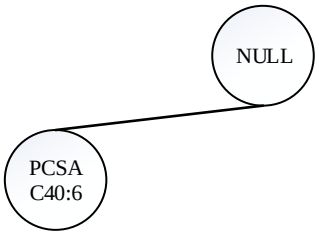
Karena *support count Item* PCSACWFF50 memenuhi *minimum support* maka *Item* PCSACWFF50 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { PCSACWFF50} akan dihilangkan dan *support count* yang satu lintasan dengan { PCSACWFF50 } akan sama dengan *support count* { PCSACWFF50 } sebelumnya.
2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



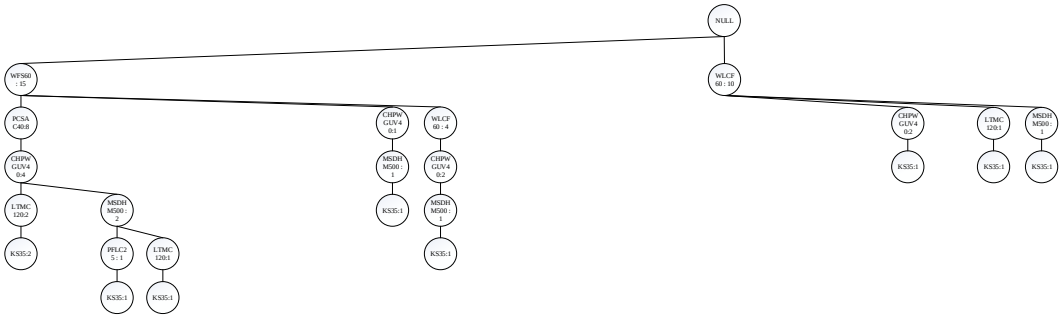
Gambar 3. 37 *Conditional FP-TreeItem* PCSACWFF50 setelah dibangkitkan

3) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { PCSAC40:6} adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 38 *Conditional FP-Tree* PCSACWFF50, PCSAC40 : 6

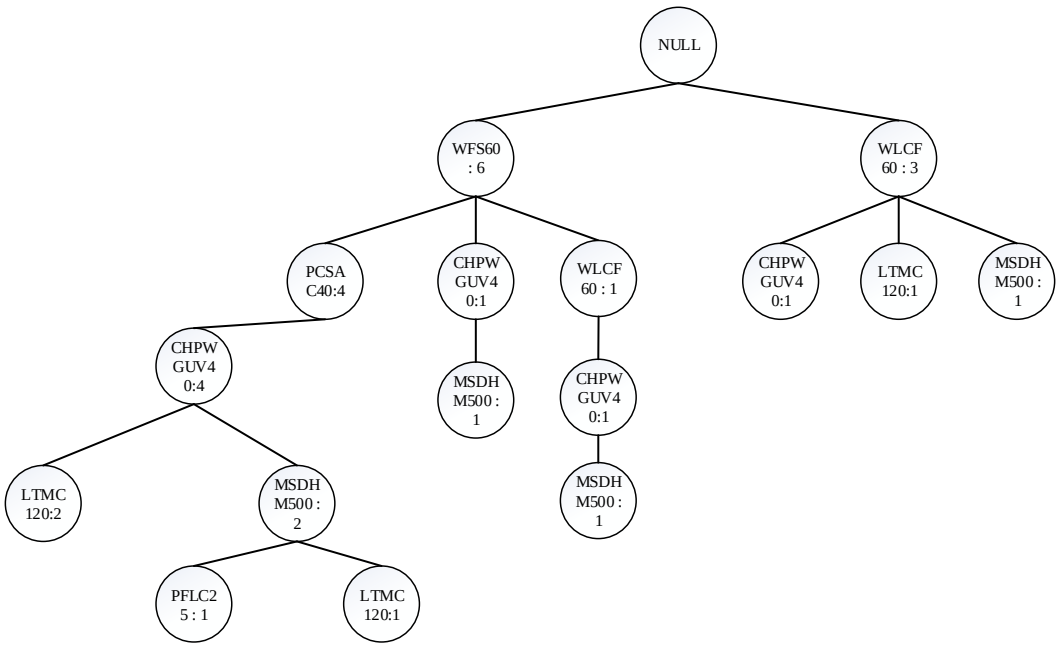
7. *Conditional FP-Tree*untuk*Item* KS35 dengan *conditional pattern base* {WFS60, PCSAC40, CHPWGUV40, LTMC120 : 2},{ WFS60, PCSAC40, CHPWGUV40, MSDHM500, PFLC25 : 1},{ WFS60, PCSAC40, CHPWGUV40, MSDHM500, LTMC120 : 1},{ WFS60, CHPWGUV40, MSDHM500 : 1},{. WFS60, WLCF60, CHPWGUV40, MSDHM500: 1},{ WLCF60, CHPWGUV40 : 1},{ WLCF60, LTMC120 : 1},{ WLCF60, MSDHM500 : 1}



Gambar 3. 39 *Conditional FP-TreeItem* KS35

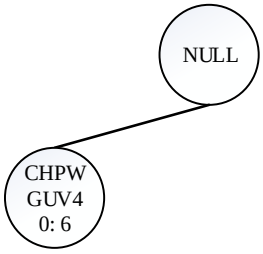
Karena *support count Item* KS35 memenuhi *minimum support* maka *Item* KS35 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { KS35} akan dihilangkan dan *support count* yang satu lintasan dengan { KS35 } akan sama dengan *support count* { KS35 } sebelumnya.
2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

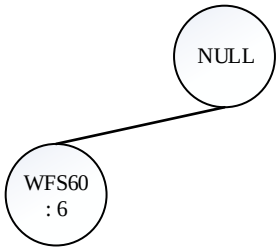


Gambar 3. 40 *Conditional FP-TreeItem* KS35 setelah dibangkitkan

3) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. {CHPWGUV40: 6},{WFS60: 6}adalah *Item* yang keluar yang dapat memenuhi minimum support.

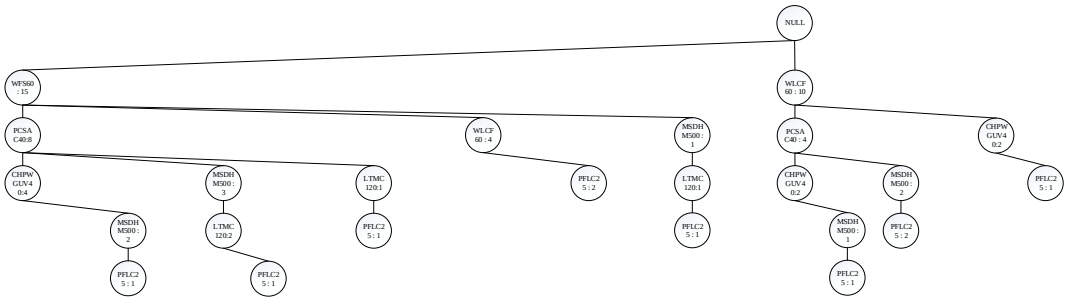


Gambar 3. 41 *Conditional FP-Tree* KS35,CHPWGUV40 : 6



Gambar 3. 42 *Conditional FP-Tree* KS35,WFS60 : 6 Hasil *Frequent pattern*

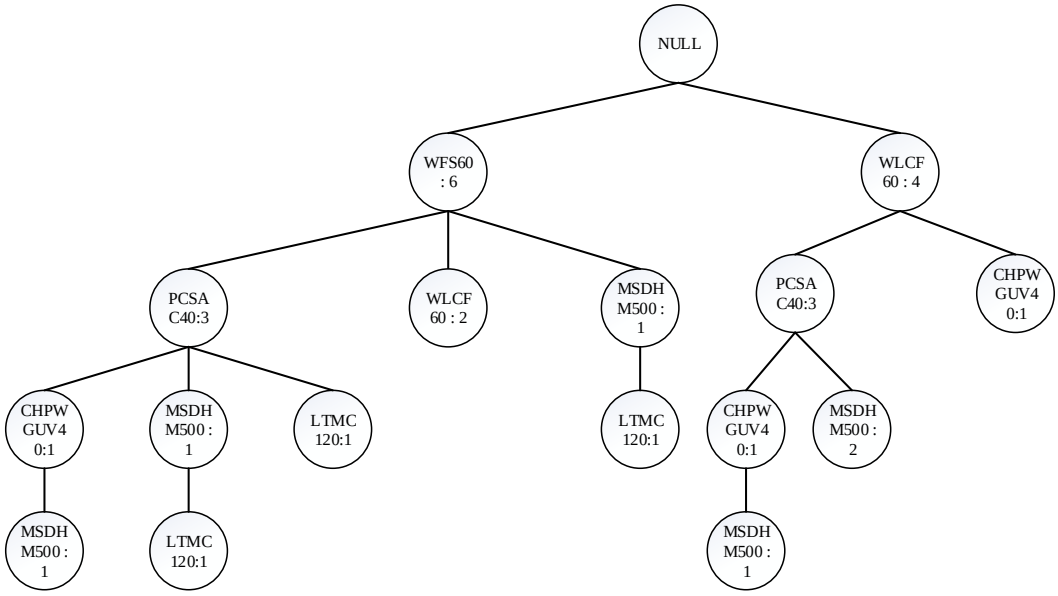
*8. Conditional FP-Tree*untuk*Item* PFLC25 dengan *conditional pattern base*



Gambar 3. 43 *Conditional FP-TreeItem* PFLC25

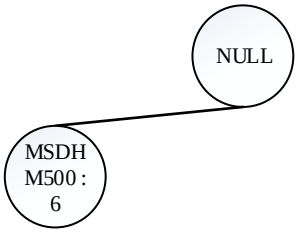
Karena *support count Item* PFLC25 memenuhi *minimum support* maka *Item* PFLC25 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { PFLC25} akan dihilangkan dan *support count* yang satu lintasan dengan { PFLC25 } akan sama dengan *support count* { PFLC25 } sebelumnya.
2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

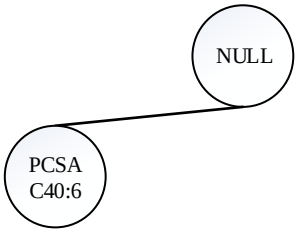


Gambar 3. 44 *Conditional FP-TreeItem* PFLC25 setelah dibangkitkan

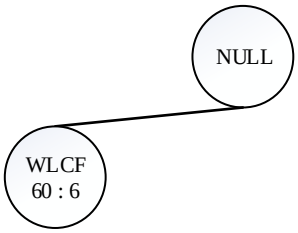
3) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. {MSDHM500 : 6},{PCSAC40:6},{WLCF60 : 6},{WFS60: 6} adalah *Item* yang keluar yang dapat memenuhi minimum support.



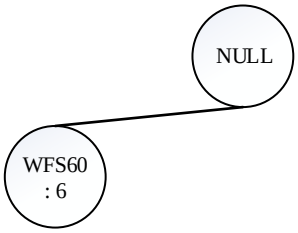
Gambar 3. 45 *Conditional FP-Tree* PFLC25, MSDHM500 : 6



Gambar 3. 46 *Conditional FP-Tree* PFLC25, PCSAC40:6

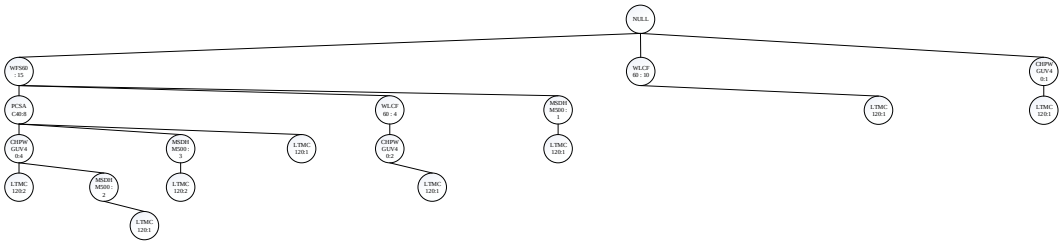


Gambar 3. 47 *Conditional FP-Tree* PFLC25, WLCF60 : 6



Gambar 3. 48 *Conditional FP-Tree* PFLC25, WFS60 : 6

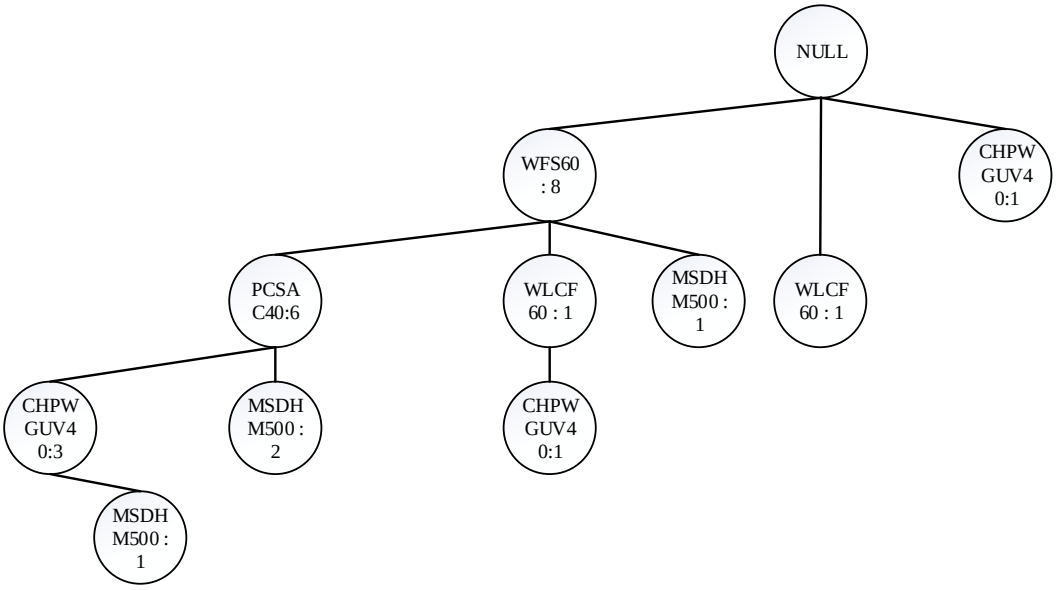
*9. Conditional FP-Tree*untuk LTMC120 *Item* dengan *conditional pattern base*



Gambar 3. 49 *Conditional FP-TreeItem* LTMC120

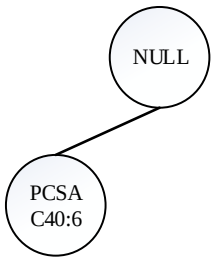
Karena *support count Item* LTMC120 memenuhi *minimum support* maka *Item* LTMC120 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

4) *Item* { LTMC120} akan dihilangkan dan *support count* yang satu lintasan dengan { LTMC120 } akan sama dengan *support count* { LTMC120 } sebelumnya.
5) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

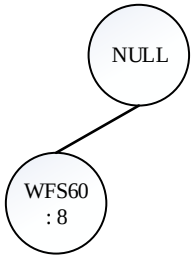


Gambar 3. 50 *Conditional FP-TreeItem* LTMC120 setelah dibangkitkan

6) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { PCSAC40:6},{WFS60: 8}adalah *Item* yang keluar yang dapat memenuhi minimum support.

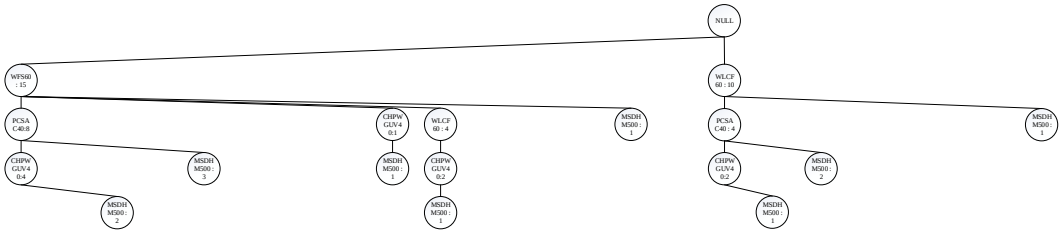


Gambar 3. 51 *Conditional FP-Tree* LTMC120, PCSAC40:6



Gambar 3. 52 *Conditional FP-Tree* LTMC120, WFS60 : 8

*10. Conditional FP-Tree*untuk MSDHM500 *Item* dengan *conditional pattern base*

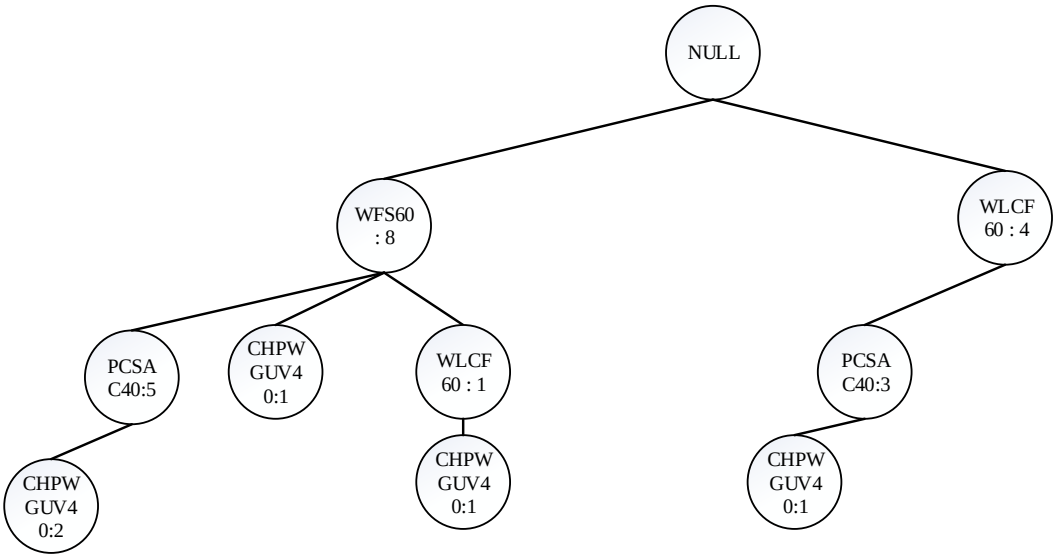


Gambar 3. 53 *Conditional FP-TreeItem* MSDHM500

Karena *support count Item* MSDHM500 memenuhi *minimum support* maka *Item* MSDHM500 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

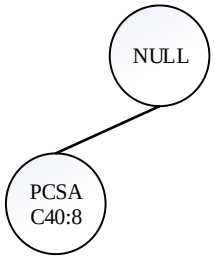
1) *Item* { MSDHM500} akan dihilangkan dan *support count* yang satu lintasan dengan { MSDHM500 } akan sama dengan *support count* { MSDHM500 } sebelumnya.

2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

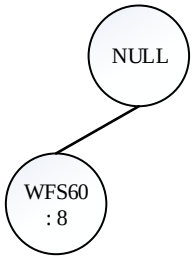


Gambar 3. 54 *Conditional FP-TreeItem* MSDHM500 setelah dibangkitkan

3) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. { PCSAC40:8},{WFS60: 8}adalah *Item* yang keluar yang dapat memenuhi minimum support.

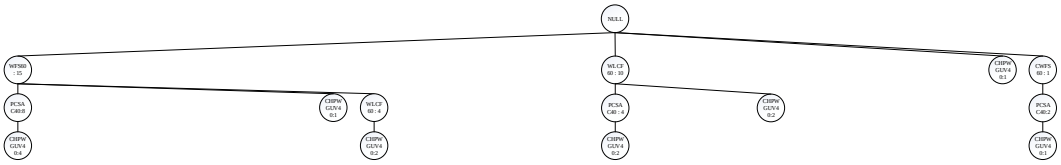


Gambar 3. 55 *Conditional FP-Tree* MSDHM500, PCSAC40:8



Gambar 3. 56 *Conditional FP-Tree* MSDHM500, WFS60 : 8

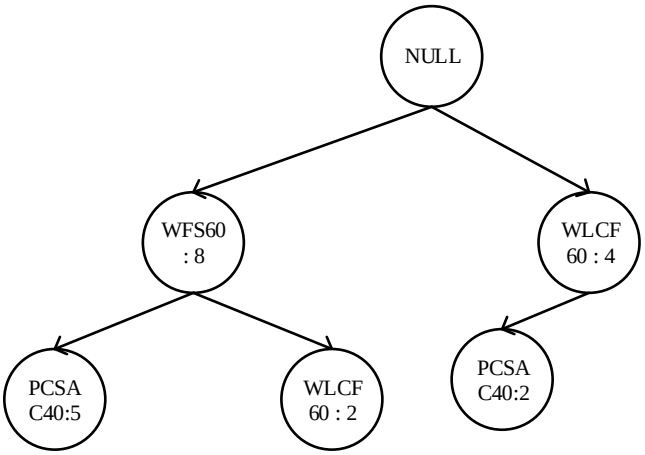
11. *Conditional FP-Tree*untuk CHPWGUV40 *Item* dengan *conditional pattern base*



Gambar 3. 57 *Conditional FP-TreeItem* CHPWGUV40

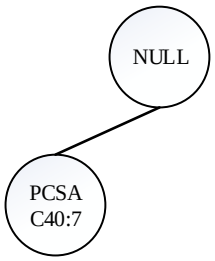
Karena *support count Item* CHPWGUV40 memenuhi *minimum support* maka *Item* CHPWGUV40 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { CHPWGUV40} akan dihilangkan dan *support count* yang satu lintasan dengan { CHPWGUV40 } akan sama dengan *support count* { CHPWGUV40 } sebelumnya.
2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.

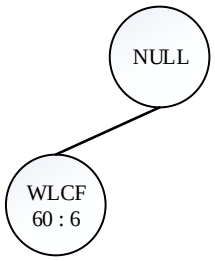


Gambar 3. 58 *Conditional FP-TreeItem* CHPWGUV40 setelah dibangkitkan

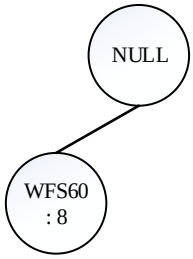
3) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*. {PCSAC40:7},{ WLCF60 : 6},{WFS60: 8}adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 59 *Conditional FP-Tree* CHPWGUV40, PCSAC40:8

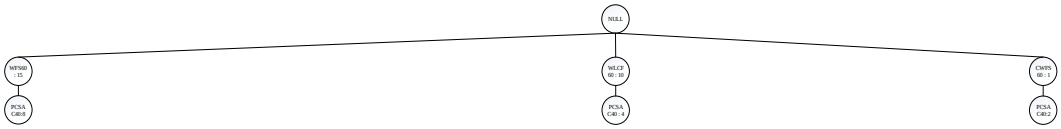


Gambar 3. 60 *Conditional FP-Tree* CHPWGUV40,WLCF60 : 6



Gambar 3. 61 *Conditional FP-Tree* CHPWGUV40, WFS60 : 8

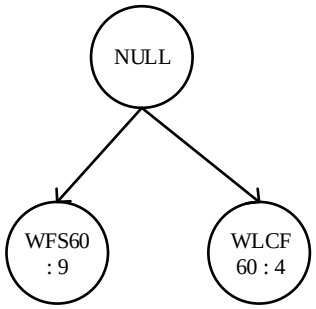
12. *Conditional FP-Tree*untuk PCSAC40 *Item* dengan *conditional pattern base*



Gambar 3. 62 *Conditional FP-TreeItem* PCSAC40

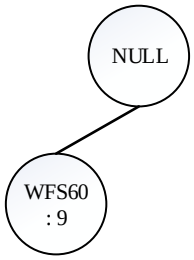
Karena *support count Item* PCSAC40 memenuhi *minimum support* maka *Item* PCSAC40 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { PCSAC40} akan dihilangkan dan *support count* yang satu lintasan dengan { PCSAC40 } akan sama dengan *support count* { PCSAC40 } sebelumnya.
2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



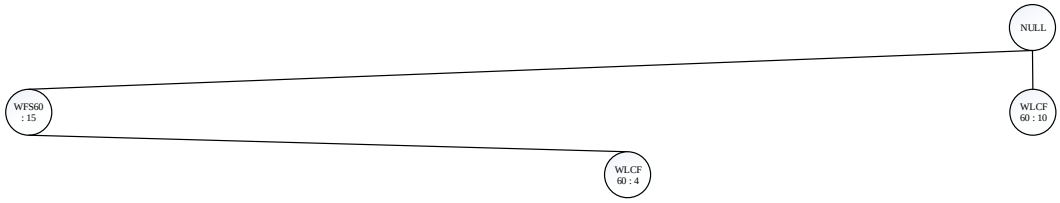
Gambar 3. 63 *Conditional FP-TreeItem* PCSAC40 setelah dibangkitkan

4) Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent pattern* yang memenuhi *minimum support*.{WFS60: 9}adalah *Item* yang keluar yang dapat memenuhi minimum support.



Gambar 3. 64 *Conditional FP-Tree* PCSAC40, WFS60 : 9

*13. Conditional FP-Tree*untuk WLCF60 *Item* dengan *conditional pattern base*

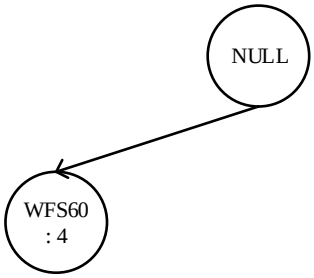


Gambar 3. 65 *Conditional FP-TreeItem* WLCF60

Karena *support count Item* WLCF60 memenuhi *minimum support* maka *Item* WLCF60 akan dibangkitkan dengan *conditional fp-tree*, berikut langkah langkahnya :

1) *Item* { WLCF60 } akan dihilangkan dan *support count* yang satu lintasan dengan { WLCF60 } akan sama dengan *support count* { WLCF60 } sebelumnya.

2) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 66 *Conditional FP-TreeItem* WLCF60 setelah dibangkitkan

*14. Conditional FP-Tree*untuk WFS60 *Item* dengan *conditional pattern base*

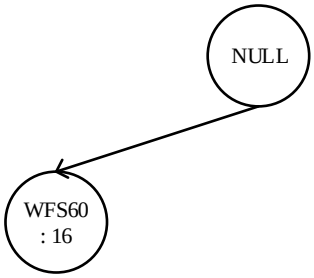


Gambar 3. 67 *Conditional FP-TreeItem* WFS60

Karena *support count Item* WFS60 memenuhi *minimum support* maka *Item* WLCF60 akan dibangkitkan dengan *conditional fp-tree,* berikut langkah langkahnya :

3) *Item* { WFS60 } akan dihilangkan dan *support count* yang satu lintasan dengan {WFS60} akan sama dengan *support count* { WFS60 } sebelumnya.

4) Apabila dalam suatu lintasan melintasi satu atau dua lintasan, maka *support count*-nya akan selalu bertambah 1.



Gambar 3. 68 *Conditional FP-TreeItem* WFS60 setelah dibangkitkan

c. Tahap Pencarian *Frequnt pattern*

Setelah mendapatkan hasil dari *conditional fp-tree*, d*item*ukan bebeapa *frequent pattern* yang keluar, dapat dilihat pada table 3.9

Tabel 3. 11 *Hasil Frequent Pattern Generated*

Item	***Frequent Pattern***
WLBBCL15	{WLBBCL15,WFS60:4},{WLBBCL15, PCSAC40:4}, {WLBBCL15, PCSACWFF50:5 }
WFSNDS	{WFSNDS ,KS35 : 4}, {WFSNDS, WLCF60 : 4}
PSHPAD170	{PSHPAD170, KS35 :6}
PELB	{PELB , WFS60:6},{PELB , CHPWGUV40 : 6},{PELB , KS35 : 6}
PAR	{PAR , WFS60:7}
PCSACWFF50	{PCSACWFF50, PCSAC40 : 6}
KS35	{KS35,CHPWGUV40 : 6},{KS35,WFS60 : 6}
PFLC25	{PFLC25, MSDHM500 : 6},{PFLC25, PCSAC40:6},{PFLC25, WLCF60 : 6} ,{PFLC25, WFS60 : 6 }
LTMC120	{LTMC120, PCSAC40:6},{LTMC120, WFS60 : 8 }
MSDHM500	{MSDHM500, PCSAC40:8},{MSDHM500, WFS60 : 8 }
CHPWGUV40	{CHPWGUV40,PCSAC40:8},{CHPWGUV40,WLCF60:6},{CHPWGUV40,WFS60:8}
PCSAC40	{PCSAC40, WFS60 : 9}
WLCF60	{WLCF60, WFS60 : 4}

Dari hasil *frequent patterns* yang didapatkan pada table 3.10 di atas akan di *generate* untuk mendapatkan *rule* dengan *minimum support* 4 dan *minimum confidence* 60 % , nilai minimum confidence ini untuk meyakinkan kuatnya hubungan antara *Item* yang satu dengan yang lain.Dengan menggunakan rumus yang ada pada bagian bab 2 untuk memperoleh nilai *support* dan nilai *confidence*.

$$Confidence = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi yang mengandung A}} \text{x}100\%..$$

Berikut hasil perhitungan *minimum confidence* dapat dilihat pada tabel 3.11 :

Tabel 3. 12 Hasil Perhitungan *Support* dan *Confidence*

Rule	***Minimum Confidence*%**

WLBBCL15 - > WFS60	4/5*100 = 80%
WFS60 -> WLBBCL15	4/16*100 = 25%
WLBBCL15 -> PCSAC40	4/5*100 = 80 %
PCSAC40 -> WLBBCL15	4/13*100 =31 %
WLBBCL15 -> PCSACWFF50	4/5*100 = 80%
PCSACWFF50 -> WLBBCL15	4/9*100 = 45 %
WFSNDS - >KS35	4/5*100 = 80 %
KS335 -> WFSNDS	4/9*100 = 44 %
WFSNDS -> WLCF60	4/5*100 = 80 %
WLCF60 -> WFSNDS	4/14*100 = 29 %
WLBBCL15 -> WFS60,PCSAC40	4/5*100% = 80 %
WFS60-> WLBBCL15,PCSAC40,PCSACWFF50	4/16*100 = 25 %
PCSAC40 -> WLBBCL15,PCSACWFF50, WFS60	4/13*100 = 31%
PCSACWFF50 -> PCSAC40, WFS60, WLBBCL15	4/9*100 = 45 %
WFSNDS -> KS35, WLCF60	4/*5*100 = 80 %
KS35 -> WFSNDS, WLCF60	4/9*100 = 44 %
WLCF60 -> WFSNDS, KS35	4/14*100 = 19 %
PSHPAD170 -> KS35	6/7*100% = 86%
KS35 -> PSHPAD170	6/9*100% = 67%
PELB -> WFS60, CHPWGUV40, KS35	6/8*100 = 75 %
WFS60 ->PELB, CHPWGUV40, KS35	6/16*100 = 37,5 %
CHPWGUV40->PELB, KS35, WFS60	6/13*100 = 46 %
KS35->PELB, WFS60, CHPWGUV40	6/9*100 = 67 %
PAR -> WFS60	7/8*100% = 88%
PCSACWFF50 -> WLBBCL15	5/9*100 = 56 %
PCSAC40 -> WFS60	9/14 * 100% = 64%
WFS60 -> PCSAC40	9/13 * 100% = 69%
CHPWGUV40 -> PCSAC40	7/13 * 100% = 54%
PCSAC40 -> CHPWGUV40	7 / 13 * 100% = 54%
CHPWGUV40, WLCF60	6/13 * 100% = 46%
WLCF60 -> CHPWGUV40	6/14 * 100 = 42%
CHPWGUV40 -> WFS60	8 / 13 * 100% = 61%
WFS60 -> CHPWGUV40	8/16 * 100 = 50%
MSDHM500, PCSAC40	8/12 * 100% = 67%

PCSA40 -> MSDHM500	8/13 * 100% = 61%
MSDHM500, WFS60	8/12 * 100% = 67%
WFS60 -> MSDHM500	8/16 * 100% = 50%
LTMC120 -> WFS60	8/10 *100% = 80%
WFS60 -> LTMC120	8/16*100% = 50%
LTMC120 -> PCSAC40	6/10*100% = 60%
WFS60 -> PAR	7/16*100% = 43%
PCSACWFF50-> PCSAC40	6/9*100% = 67%
PCSAC40 -> PCSACWFF50	6/13*100 = 46%
KS35 -> CHPWGUV40, WFS60	6/9*100 = 67 %
CHPWGUV40 -> KS35, WFS60	6/13*100 = 46 %
WFS60 -> KS35, CHPWGUV40	6/16*100 = 37.5 %
PFLC25 -> MSDHM500, PCSAC40, WLCF60, WFS60	6/10*100 = 60 %
MSDHM500 ->PFLC25, PCSAC40, WLCF60, WFS60	6/12*100 = 50 %
PCSAC40 -> PFLC25, WLCF60, WFS60, MSDHM500	6/13*100 = 46 %
WLCF60 ->PFLC25, WFS60, MSDHM500, PCSAC40	6/14*100 = 42 %
WFS60 ->PFLC25, MSDHM500, PCSAC40, WLCF60	6/16*100 = 37.5 %
LTMC120 -> PCSAC40, WFS60	8/10*100 = 80 %
PCSAC40 -> LTMC120, WFS60	8/13*100 = 62 %
WFS60 -> LTMC120, PCSAC40	8/16*100 = 50 %
MSDHM500 -> PCSAC40, WFS60	8/12*100 = 67 %
PCSAC40 -> MSDHM500, WFS60	8/13*100 = 62 %
WFS60 -> MSDHM500, PCSAC40	8/16*100 = 50 %
PCSAC40 -> LTMC120	6/13*100% =46%
PFLC25 -> MSDHM500	6/10*100% = 60%
MSDHM500 -> PFLC25	6/12*100% =50%
PFLC25 -> PCSAC40	6/10*100% = 60%
PCSAC40 -> PFLC25	6/13*100 = 46%
PFLC25 -> WLCF60	6/10*100% = 60%
WLCF60 -> PFLC25	6/14*100% = 42%
PFLC25 -> WFS60	6/10*100% = 60%
WFS60 -> PFLC25	6/16*100% = 38%
KS35 -> CHPWGUV40	6/9*100% = 67%
CHPWGUV40 -> KS35	6/13*100% = 46%
KS35 -> WFS60	6/9*100% = 67%

WFS60 -> KS35	6/16*100% = 38%
PCSACWFF50-> PCSAC40	6/9*100% = 67%
PCSAC40 -> PCSACWFF50	6/13*100 = 46%
CHPWGUV40 -> PCSAC40, WLCF60, WFS60	813*100 = 62 %
PCSAC40 -> CHPWGUV40, WLCF60, WFS60	8/13*100 = 62 %
WLCF60-> CHPWGUV40, WFS60, PCSAC40	8/14*100 = 57 %
WFS60-> CHPWGUV40, PCSAC40, WLCF60	8/16*100 = 50 %
PCSAC40 -> WFS60	9/14 * 100% = 64%
WFS60 -> PCSAC40	9/13 * 100% = 69%
PAR -> WFS60	7/8*100% = 88%
WFS60 -> PAR	7/16*100% = 43%
PELB -> KS35	6/8*100% = 75%
KS35 -> PELB	6/9*100% = 67%
PELB -> WFS60	6/8*100% = 75%
WFS60 -> PELB	6/16*100% = 38%
PELB -> CHPWGUV40	6/8*100% = 75%
CHWPGUV40 -> PELB	6/13*100% = 46%
PSHPAD170 -> KS35	6/6*100% = 100%
KS35 -> PSHPAD170	6/9*100% = 67%

Setelah melakukan perhitungan *minimum support* dan *minimum confidence* pada table 3.11, proses selanjutnya adalah mencari *generate rule* yang memenuhi nilai *minimum support* dan *confidence* yang telah ditentukan sebelumnya. Jika terdapat *Itemset* yang tidak memenuhi *support* dan *confidence* akan dihilangkan atau dihapus. Hasil *generate* dapat dilihat pada tabel 3.12

Tabel 3. 13hasil *Generate Rule*

Rule	***Minimum Confidence*%**
WLBBCL15 - > WFS60	4/5*100 = 80%
WLBBCL15 -> PCSAC40	4/5*100 = 80 %
WLBBCL15 -> PCSACWFF50	4/5*100 = 80%
WFSNDS - >KS35	4/5*100 = 80 %
WFSNDS -> WLCF60	4/5*100 = 80 %
WLBBCL15 -> WFS60,PCSAC40	4/5*100% = 80 %
WFSNDS -> KS35, WLCF60	4/*5*100 = 80 %

PSHPAD170 -> KS35	6/7*100% = 86%
KS35 -> PSHPAD170	6/9*100% = 67%
PELB -> WFS60, CHPWGUV40, KS35	6/8*100 = 75 %
KS35->PELB, WFS60, CHPWGUV40	6/9*100 = 67 %
PAR -> WFS60	7/8*100% = 88%
PCSAC40 -> WFS60	9/14 * 100% = 64%
WFS60 -> PCSAC40	9/13 * 100% = 69%
CHPWGUV40 -> WFS60	8 / 13 * 100% = 61%
WFS60 -> CHPWGUV40	8/16 * 100 = 50%
MSDHM500, PCSAC40	8/12 * 100% = 67%
PCSA40 -> MSDHM500	8/13 * 100% = 61%
MSDHM500, WFS60	8/12 * 100% = 67%
LTMC120 -> WFS60	8/10 *100% = 80%
LTMC120 -> PCSAC40	6/10*100% = 60%
PCSACWFF50-> PCSAC40	6/9*100% = 67%
KS35 -> CHPWGUV40, WFS60	6/9*100 = 67 %
PFLC25 -> MSDHM500, PCSAC40, WLCF60, WFS60	6/10*100 = 60 %
LTMC120 -> PCSAC40, WFS60	8/10*100 = 80 %
PCSAC40 -> LTMC120, WFS60	8/13*100 = 62 %
WFS60 -> LTMC120, PCSAC40	8/16*100 = 50 %
MSDHM500 -> PCSAC40, WFS60	8/12*100 = 67 %
PCSAC40 -> MSDHM500, WFS60	8/13*100 = 62 %
PFLC25 -> MSDHM500	6/10*100% = 60%
PFLC25 -> PCSAC40	6/10*100% = 60%
PFLC25 -> WLCF60	6/10*100% = 60%
PFLC25 -> WFS60	6/10*100% = 60%
KS35 -> CHPWGUV40	6/9*100% = 67%
KS35 -> WFS60	6/9*100% = 67%
PCSACWFF50-> PCSAC40	6/9*100% = 67%
CHPWGUV40 -> PCSAC40, WLCF60, WFS60	813*100 = 62 %
PCSAC40 -> CHPWGUV40, WLCF60, WFS60	8/13*100 = 62 %
PCSAC40 -> WFS60	9/14 * 100% = 64%
WFS60 -> PCSAC40	9/13 * 100% = 69%
PAR -> WFS60	7/8*100% = 88%
PELB -> KS35	6/8*100% = 75%
KS35 -> PELB	6/9*100% = 67%

PELB -> WFS60	6/8*100% = 75%
PELB -> CHPWGUV40	6/8*100% = 75%
PSHPAD170 -> KS35	6/6*100% = 100%
KS35 -> PSHPAD170	6/9*100% = 67%

Rule yang telah memenuhi nilai *minimum confidance* kemudian disesuaikan dengan nama barang. Sehingga kode barang yang terdapat pada *rule* sesuai dengan nama barang dan dapat dijadikan informasi. Tabelnama barang dapat dilihat pada tabel 3.13:

Tabel 3. 14 Nama Barang

KODE	NAMA PRODUK
WFS60	Wardah Facial Scrub 60 ml
WLCF60	WARDAH Light Creamy Foam 60 ML
PCSAC40	PONDS CS ACNE C 40
CHPWGUV40	CITRA HAZ SPOTLES WHITE GLOW UV 40 G
MSDHM500	MAKARIZO SALON DAILY HAIR MASK 500gr NEW
LTMC120	La Tulip Milk Cleanser 120 ml New
PFLC25	PONDS FAIR LOVELY CREAM 25 G
KS35	KAPAS SARIAYU 35
PCSACWFF50	PONDS CS ACNE CLEAR WHITE FF50 G
PAR	Pensil Alis Revlon
PELB	Pixy Eye Liner Black
PSHPAD170	PANTENE SHP ANTI DANRUFF 170 ML ATLAS
WLDCS30S1	Wardah Lig Day Cream Step 1 30 gr
WLNC20S2	Wardah LIG NIGHT CREAM 20 step 2
WFSNDS	WARDAH FACIAL SERUM NORMAL DRY SKIN
WLBBCL15	WARDAH LIGHT BB CREAM LIGHT 15ML

Informasi yang dihasilkan berupa kombinasi *Item* dan nama barang berdasarkan kode barangnya sesuai dengan tabel 3.13. dapat dilihat pada tabel 3.14 :

Tabel 3. 15 Informasiyang dihasilkan dari *rule*

Informasi Rekomendasi Paket	Confidence
CITRA HAZ SPOTLES WHITE GLOW UV 40 G & Wardah Facial Scrub 60 ml	62
PONDS CS ACNE C 40 & Wardah Facial Scrub 60 ml	69

MAKARIZO SALON DAILY HAIR MASK 500gr NEW & Wardah Facial Scrub 60 ml	67
MAKARIZO SALON DAILY HAIR MASK 500gr NEW & PONDS CS ACNE C 40	67
La Tulip Milk Cleanser 120 ml New & Wardah Facial Scrub 60 ml	80
La Tulip Milk Cleanser 120 ml New & PONDS CS ACNE C 40	60
La Tulip Milk Cleanser 120 ml New & PONDS CS ACNE C 40& Wardah Facial Scrub 60 ml	60
PONDS FAIR LOVELY CREAM 25 G & Wardah Facial Scrub 60 ml	60
PONDS FAIR LOVELY CREAM 25 G & WARDAH Light Creamy Foam 60 ML	60
PONDS FAIR LOVELY CREAM 25 G & PONDS CS ACNE C 40	60
PONDS FAIR LOVELY CREAM 25 G & MAKARIZO SALON DAILY HAIR MASK 500gr NEW	60
KAPAS SARIAYU 35 & Wardah Facial Scrub 60 ml	67
KAPAS SARIAYU 35 & CITRA HAZ SPOTLES WHITE GLOW UV 40 G	78
KAPAS SARIAYU 35 & CITRA HAZ SPOTLES WHITE GLOW UV 40 G& Wardah Facial Scrub 60 ml	67
PONDS CS ACNE CLEAR WHITE FF50 G & PONDS CS ACNE C 40	67
Pensil Alis Revlon & Wardah Facial Scrub 60 ml	88
Pensil Alis Revlon & CITRA HAZ SPOTLES WHITE GLOW UV 40 G	62
Pixy Eye Liner Black & Wardah Facial Scrub 60 ml	75
Pixy Eye Liner Black & CITRA HAZ SPOTLES WHITE GLOW UV 40 G	88
Pixy Eye Liner Black & CITRA HAZ SPOTLES WHITE GLOW UV 40 G & Wardah Facial Scrub 60 ml	62
Pixy Eye Liner Black & PONDS CS ACNE C 40	62
Pixy Eye Liner Black & MAKARIZO SALON DAILY HAIR MASK 500gr NEW	62
Pixy Eye Liner Black & MAKARIZO SALON DAILY HAIR MASK 500gr NEW& Wardah Facial Scrub 60 ml	62
Pixy Eye Liner Black & KAPAS SARIAYU 35	75
Pixy Eye Liner Black & KAPAS SARIAYU 35 & Wardah Facial Scrub 60 ml	62
Pixy Eye Liner Black & KAPAS SARIAYU 35& CITRA HAZ SPOTLES WHITE GLOW UV 40 G	75
Pixy Eye Liner Black & KAPAS SARIAYU 35& CITRA HAZ SPOTLES WHITE GLOW UV 40 G& Wardah Facial Scrub 60 ml	62
PANTENE SHP ANTI DANRUFF 170 ML ATLAS & CITRA HAZ SPOTLES WHITE GLOW UV 40 G	71
PANTENE SHP ANTI DANRUFF 170 ML ATLAS & KAPAS SARIAYU 35	86
Wardah Lig Day Cream Step 1 30 gr & Wardah Facial Scrub 60 ml	67
Wardah Lig Day Cream Step 1 30 gr & WARDAH Light Creamy Foam 60 ML	67

Wardah Lig Day Cream Step 1 30 gr & CITRA HAZ SPOTLES WHITE GLOW UV 40 G	67
Wardah Lig Day Cream Step 1 30 gr & La Tulip Milk Cleanser 120 ml New	67
Wardah Lig Day Cream Step 1 30 gr & KAPAS SARIAYU 35	67
Wardah Lig Day Cream Step 1 30 gr & Pensil Alis Revlon	67
Wardah LIG NIGHT CREAM 20 step 2 & WARDAH Light Creamy Foam 60 ML	67
Wardah LIG NIGHT CREAM 20 step 2 & MAKARIZO SALON DAILY HAIR MASK 500gr NEW	67
WARDAH FACIAL SERUM NORMAL DRY SKIN & WARDAH Light Creamy Foam 60 ML	80
WARDAH FACIAL SERUM NORMAL DRY SKIN & KAPAS SARIAYU 35	100
WARDAH FACIAL SERUM NORMAL DRY SKIN & KAPAS SARIAYU 35& WARDAH Light Creamy Foam 60 ML	80
WARDAH LIGHT BB CREAM LIGHT 15ML & Wardah Facial Scrub 60 ml	80
WARDAH LIGHT BB CREAM LIGHT 15ML & PONDS CS ACNE C 40	80
WARDAH LIGHT BB CREAM LIGHT 15ML & PONDS CS ACNE CLEAR WHITE FF50 G	100
WARDAH LIGHT BB CREAM LIGHT 15ML & PONDS CS ACNE CLEAR WHITE FF50 G& Wardah Facial Scrub 60 ml	80
WARDAH LIGHT BB CREAM LIGHT 15ML & PONDS CS ACNE CLEAR WHITE FF50 G& PONDS CS ACNE C 40	80

4. Aturan Bisnis Pembentukan Paket

Dalam menentukan paket yang akan ditawarkan pihak Makmur Jaya Kosmetik memiliki pertimbangan yaitu :

a. Jumlah Produk yang di paketkan yaitu sebanyak 3.

b. Harga minimal produk Rp.30.000 s/d Rp.150.000.

Tabel 3. 16 Tabel Rekomendasi Paket

PAKET	PRODUK	HARGA
PAKET 1	CITRA HAZ SPOTLES WHITE GLOW UV 40 G, Wardah Facial Scrub 60 ml	Rp 49500
PAKET 2	PONDS CS ACNE C 40, Wardah Facial Scrub 60 ml	Rp 61500
PAKET 3	MAKARIZO SALON DAILY HAIR MASK 500gr NEW, Wardah Facial Scrub 60 ml	Rp 114000
PAKET 4	MAKARIZO SALON DAILY HAIR MASK 500gr NEW, PONDS CS ACNE C 40	Rp 97500
PAKET 5	La Tulip Milk Cleanser 120 ml New, Wardah Facial Scrub 60 ml	Rp 62000

PAKET 6	La Tulip Milk Cleanser 120 ml New, PONDS CS ACNE C 40	Rp 45500
PAKET 7	La Tulip Milk Cleanser 120 ml New, PONDS CS ACNE C 40, Wardah Facial Scrub 60 ml	Rp 84500
PAKET 8	PONDS FAIR LOVELY CREAM 25 G, Wardah Facial Scrub 60 ml	Rp 50500
PAKET 9	PONDS FAIR LOVELY CREAM 25 G, WARDAH Light Creamy Foam 60 ML, WARDAH Light Creamy Foam 60 ML	Rp 109500
PAKET 10	PONDS FAIR LOVELY CREAM 25 G, PONDS CS ACNE C 40	Rp 34000
PAKET 11	PONDS FAIR LOVELY CREAM 25 G, MAKARIZO SALON DAILY HAIR MASK 500gr NEW	Rp 86500
PAKET 12	KAPAS SARIAYU 35, Wardah Facial Scrub 60 ml	Rp 44000
PAKET 13	KAPAS SARIAYU 35, CITRA HAZ SPOTLES WHITE GLOW UV 40 G, Wardah Facial Scrub 60 ml	Rp 54500
PAKET 14	PONDS CS ACNE CLEAR WHITE FF50 G, PONDS CS ACNE C 40	Rp 46500
PAKET 15	Pensil Alis Revlon, Wardah Facial Scrub 60 ml	Rp 44000
PAKET 16	Pixy Eye Liner Black, Wardah Facial Scrub 60 ml	Rp 67000
PAKET 17	Pixy Eye Liner Black, CITRA HAZ SPOTLES WHITE GLOW UV 40 G	Rp 38500
PAKET 18	Pixy Eye Liner Black, CITRA HAZ SPOTLES WHITE GLOW UV 40 G, Wardah Facial Scrub 60 ml	Rp 77500
PAKET 19	Pixy Eye Liner Black, PONDS CS ACNE C 40	Rp 50500
PAKET 20	Pixy Eye Liner Black, MAKARIZO SALON DAILY HAIR MASK 500gr NEW	Rp 103000
PAKET 21	Pixy Eye Liner Black, MAKARIZO SALON DAILY HAIR MASK 500gr NEW, Wardah Facial Scrub 60 ml	Rp 142000
PAKET 22	Pixy Eye Liner Black, KAPAS SARIAYU 35	Rp 33000
PAKET 23	Pixy Eye Liner Black, KAPAS SARIAYU 35, Wardah Facial Scrub 60 ml	Rp 72000
PAKET 24	Pixy Eye Liner Black, KAPAS SARIAYU 35, CITRA HAZ SPOTLES WHITE GLOW UV 40 G	Rp 43500
PAKET 25	Pixy Eye Liner Black, KAPAS SARIAYU 35, CITRA HAZ SPOTLES WHITE GLOW UV 40 G, Wardah Facial Scrub 60 ml	Rp 82500
PAKET 26	PANTENE SHP ANTI DANRUFF 170 ML ATLAS, CITRA HAZ SPOTLES WHITE GLOW UV 40 G	Rp 32500
PAKET 27	Wardah Lig Day Cream Step 1 30 gr, Wardah Facial Scrub 60 ml	Rp 63000
PAKET 28	Wardah Lig Day Cream Step 1 30 gr, WARDAH Light Creamy Foam 60 ML, WARDAH Light Creamy Foam 60 ML	Rp 122000
PAKET 29	Wardah Lig Day Cream Step 1 30 gr, CITRA HAZ SPOTLES WHITE GLOW UV 40 G	Rp 34500
PAKET 30	Wardah Lig Day Cream Step 1 30 gr, La Tulip Milk Cleanser 120 ml New	Rp 47000

PAKET 31	Wardah LIG NIGHT CREAM 20 step 2, WARDAH Light Creamy Foam 60 ML, WARDAH Light Creamy Foam 60 ML	Rp 137000
PAKET 32	Wardah LIG NIGHT CREAM 20 step 2, MAKARIZO SALON DAILY HAIR MASK 500gr NEW	Rp 114000
PAKET 33	WARDAH FACIAL SERUM NORMAL DRY SKIN, WARDAH Light Creamy Foam 60 ML, WARDAH Light Creamy Foam 60 ML	Rp 150000
PAKET 34	WARDAH FACIAL SERUM NORMAL DRY SKIN, KAPAS SARIAYU 35	Rp 57000
PAKET 35	WARDAH LIGHT BB CREAM LIGHT 15ML, Wardah Facial Scrub 60 ml	Rp 88000
PAKET 36	WARDAH LIGHT BB CREAM LIGHT 15ML, PONDS CS ACNE C 40	Rp 71500
PAKET 37	WARDAH LIGHT BB CREAM LIGHT 15ML, PONDS CS ACNE CLEAR WHITE FF50 G	Rp 73000
PAKET 38	WARDAH LIGHT BB CREAM LIGHT 15ML, PONDS CS ACNE CLEAR WHITE FF50 G, Wardah Facial Scrub 60 ml	Rp 112000
PAKET 39	WARDAH LIGHT BB CREAM LIGHT 15ML, PONDS CS ACNE CLEAR WHITE FF50 G, PONDS CS ACNE C 40	Rp 95500

3.1.3 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Analisis kebutuhan 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

User atau pengguna yang mengoperasikan aplikasi ini merupakan manager operasional di Makmur Jaya Kosmetik. Analisa penguna dapat dilihat pada tabel 3.17.

Tabel 3. 17 *Analisis Pengguna*

User	Akses	Keahlian
Manager Operasional	Dapat mengolah data, seperti: data transaksi yang nantinya akan diproses pada aplikasi data *mining* untuk memperoleh informasi.	Dapat menginputkan data transaksi, penghapusan data transaksi dan juga dapat mengolah informasi yang

		didapat dari aplikasi data *mining*.

3.1.3 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak adalah kebutuhan kebutuhan apa saja yang diperlukan untuk membangun sebuah perangkat lunak sesuai dengan kebutuhan penggunanya. Spesifikasi kebutuhan perangkat lunak dibagi menjadi dua yaitu kebuthan *non-fungsional* dan kebutuhan *fungsional*.

3.1.3.1 Analisis kebutuhan Non-Fungsional

Analisis kebutuhan *non fungsional* merupakan bagian penting lainnya agar simulasi perangkat lunak dapat berjalan dengan baik serta untuk mengetahui elemen-elemen yang berhubungan dengan sistem yang sedang berjalan. Tahapan analisis kebutuhan *non fungsional* dibagi menjadi beberapa tahap yaitu analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak dan analisis kebutuhan pengguna (*user*).

1. Analisis Kebutuhan Perangkat Keras

Perangkat keras pendukung dalam pembangunan perangkat lunak ini adalah sebagai berikut :

a. Analisis kebutuhan perangkat keras pada sistem yang sedang berjalan di Makmur Jaya Kosmetik sendiri terdapat dua buah komputer dengan spesifikasi sebagai berikut, dapat dilihat pada tabel 3.17 :

Tabel 3. 18 Spesifikasi Perangkat Keras Yang Sedang Berjalan

No	Perangkat Keras	Spesifikasi
1	RAM	2 GB
2	Harddisk	256 GB
3	Monitor	LCD Touchscreen 16 inchi
4	Processor	Intel Pentium Core 2 Duo 2,6 Ghz

b. Analisis kebutuhan perangkat keras pada sistem yang akan dibangun membutuhkan spesifikasi *minimum* perangkat keras sebagai berikut, dapat dilihat pada tabel 3.18:

Tabel 3. 19 Spesifikasi Perangkat Keras Yang Akan Dibangun

No	Perangkat Keras	Spesifikasi
1	RAM	512 MB
2	Harddisk	1 GB
3	Monitor	LCD 16 inchi
4	Processor	Berkecepatan minimal 1,9 Ghz

c. Evaluasi kebutuhan perangkat keras

Dengan membandingkan spesifikasi perangkat keras untuk menjalankan perangkat keras yang akan dibangun dengan spesifikasi perangkat keras yang ada pada Makmur Jaya Kosmetik, maka dapat disimpulkan bahwa perangkat keras yang tersedia sesuai dengan kebutuhan dan telah memenuhi spesifikasi seperti pada tabel 3.17 diatas.

2. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak pendukung dalam pembangunan perangkat lunak ini adalah

a. Analisis kebutuhan perangkat lunak pada sistem yang sedang berjalan di Makmur Jaya Kosmetik adalah sebagai berikut, dapat dilihat pada tabel 3.19:

Tabel 3. 20 Spesifikasi Perangkat Lunak yang Sedang Berjalan

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7

b. Analisis kebutuhan perangkat lunak pada sistem yang dibangun adalah sebagai berikut, pada dilihat pada tebel 3.20 :

Tabel 3. 21 Spesifikasi Perangkat Lunak yang Akan Dibangun

No	Perangkat Keras	Spesifikasi

1	Sistem Operasi	Microsoft Windows 7
2	*Web Server* Pembangun	Xampp Server
3	DBMS	MySQL
4	*Compiler*	Microsoft Visual Studio

c. Evaluasi kebutuhan perangkat lunak

Dengan membandingkan spesifikasi perangakat lunak yang dibutuhkan dengan perangkat lunak yang ada pada Makmur Jaya Kosmetik, dapat disimpulkan bahwa spesifikasi perangkat lunak yang ada di Makmur Jaya Kosmetik masih belum memenuhi kebetuhan yang diperlukan karena pada perangkat lunak tidak terdapat *database server*, sehingga perlu ditambahkan *xampp server*.

3. Analisis Pengguna (*User*)

Analisis kebutuhan perangkat pikir digunakan untuk menganalisa karakteristik pengguna perangkat lunak yang akan dibangun. Adapun beberapa karakteristiknya adalah sebagai berikut :

Tabel 3. 22 Analisa Kebutuhan di Makmur Jaya Kosmetik

User	Tanggung Jawab	Kealian
Manager	Merencanakan,melaksanakan, dan mengevaluasi terhadap strategi bisnis yang digunakan	Mampu mengoperasikan komputer

Adapun daftar pengguna yang dibutuhkan dalam menjalankan perangkat lunak yang akan dibangun adalah sebagai berikut :

Tabel 3. 23 Analisis Kebutuhan Pada Sistem

User	Hak Akses	Keahlian
Manager	Manager dapat mengolah data transaksi penjualan dan memperoleh informasi dari aplikasi *mining*.	Pengoperasian komputer dan pemahaman konten yang tersedia pada program

3.1.3.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional yang akan dibuat meliputi Diagram *use case*, use case skenario, Diagram aktivity, Diagram *class* dan Diagram sequens.

1. *Diagram Use Case*

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan(behavior) sistem yang akan dibuat. Berikut adalah tabel definisi *user*

Tabel 3. 24 Tabel Definisi User

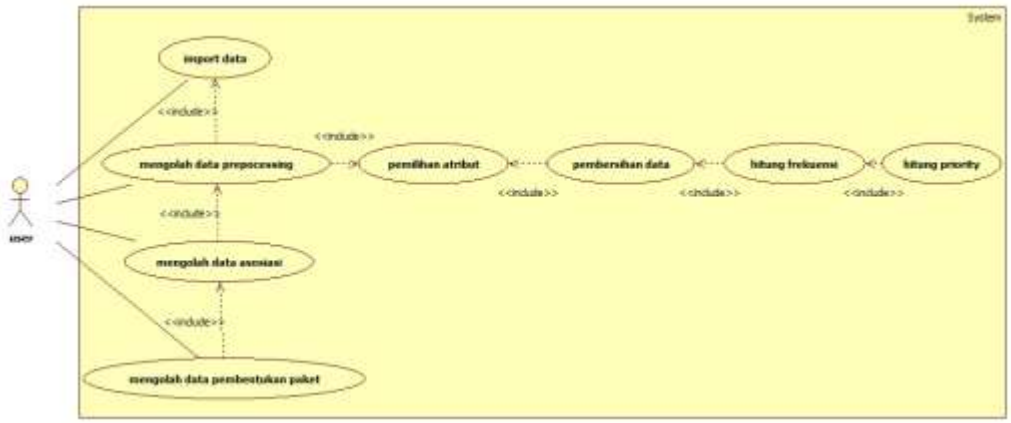
User	**Deskripsi**
User	*User* bertugas untuk melakukan *import* data, memasukkan *Minimum support* dan *Minimum Confidence*

Tabel 3. 25 Tabel Deskripsi User

No	***Use Case***	**Deskripsi**
1	*Import* Data	Import data harus dilakukan user terlebih dahulu, data yang diimport merupakan data transaksi yang selanjutnya data tersebut akan disimpan ke dalam database
2	*Preprocessing*	Data transaksi yang sudah tersimpan dalam database kemudian akan dilakukan proses pembersihan data , pemilihan atribut, dan hitung frekuensi
3	Pemilihan Atribut	Sistem melakukan pemilihan atribut Bon dan Nama Barang, proses ini hanya bisa dilakukan jika user sudah pembersihan data
4	Pembersihan Data	Sistem melakukan pembersihan data pada Bon yang mengandung Nama Barang tunggal, proses ini hanya bisa dilakukan jika user sudah melakukan *import* data
5	Hitung *frekuensi*	Sistem melakukan proses perhitungan kemunculan tiap *item* pada data transaksi, proses ini hanya bias dilakukan jika user sudah melakukan import data, pemilian atribut dan pembersihan data
6	Priority	Sistem melakukan pengurutan berdasarkan *priority* dari data transaksi yang dimportt oleh user
7	Asosiasi	*User* menginputkan nilai parameter yaitu nilai *Minimum support* dan *Minimum Confidence*, kemudian sistem melakukan pencarian *frequent itemset*, melakukan proses

No	*Use Case*	Deskripsi
		eliminasi, menghasilkan *rule* dan informasi rekomendasi paket.
9	Lihat *rules*	Sistem menampilkan hasil dari seleksi *rule*.
10	Lihat Informasi Paket	Sistem menampilkan informasi berupa produk apa saja yang bisa dipasangkan dalam bentuk paket.

Adapun Diagram *use case* dari sistem yang akan dibuat dapat dilihat di



Gambar 3. 69 Usecase Diagram

2. *Use Case Scenario*

1) Use Case Scenario Import Data

Skenario *use case import* data menggambarkan langkah – langkah aksi *user* terhadap sistem untuk melakukan *import* data transaksi penjualan yang akan disimpan ke dalam *database.* Skenario use case import data dapat dilihat pada tabel 3.25.

Tabel 3. 26 Use Case Scenario Import Data

Use case Name	**Import Data**
Goal In Context	Import data transaksi dan menyimpan ke dalam database
Precondition	*User* berada pada *form import* data
Successful End Condition	Data berhasil disimpan ke dalam *database*
Failed End Condition	Data nilai tidak berhasil disimpan ke dalam *database*
Actors	Manager

Triger	*User* memilih menu *import* data	
Main Flow	Step	Action
	1	User mencari file csv
	2	User menyimpan file
	3	Sistem menyimpan file ke *database*
	4	Sistem menampilkan data file yang dimasukan
Extension	Step	Branching Action
	1.1	*User* membatalkan proses *import*
	1.2	Sistem menampilkan pesan data tidak berhasil diimport

2) *Use Case ScenarioPreprocesing*

Skenario *use case preprocessing* menjelaskan langkah-langkah untuk melakukan proses clean data dan selected atribut. Skenario *use case preprocesing* dapat dilihat pada Tabel 3.26.

Tabel 3. 27 Skenario *Use Case Prepocessing*

Use case Name	***Preprocesing***	
Goal In Context	Sistem melakukan proses pemilihan atribut data, pembersihan data, perhitungan frekuensi, priority	
Precondition	Data transaksi telah disimpan didalam *database*	
Successful End Condition	Sistem berhasil melakukan proses pemilihan atribut data, pembersihan data, perhitungan frekuensi, priority	
Failed End Condition	Sistem tidak berhasil melakukan pemilihan atribut data, pembersihan data, perhitungan frekuensi, priority	
Actors	Manager	
Triger	*User* melakukan *request* proses pemilihan atribut dan pembersihan data kepada sistem	
Included Cases	Pemilihan atribut data, Pembersihan data, Perhitungan Frekuensi, Priority	
Main Flow	Step	Action
	1	User memilih menu preprocessing
	2	User melakukan *request*pemilihan atribut
	3	Sistem melakukan sleksi atribut
	4	Sistem menampilkan data pemilihan atribut
	5	User melakakukan *request* pembersihan data
	6	Sistem melakukan proses pembersihan data
	7	Sistem menampilkan hasil pembersihan
	8	User melakukan *request* frekeunsi
	9	Sistem melakukan perhitungan pada data yang telah dicelaning
	10	Sistem menampilkan data hasil perhitungan frekuensi
	11	User melakukan *request priority*
	12	Sistem melakukan pengurtutan berdasarkan *priority*

	13	Sistem menampilkan hasil pengurutan berdasarkan *priority*
Extension	Step	Branching Action
	-	-

3) *Use Case Scenario* Pemilihan Atribut

Skenario *use case seleksi atribut* menggambarkan langkah-langkah melakukan pemilihan atribut terhadap data transaksi yang telah melewati proses *import data*. Skenario *use case seleksi atribut* dapat dilihat pada table 3.27.

Tabel 3. 28 Use Case Scenario Pemilihan Atribut

Use case Name	**Seleksi Atribut**	
Goal In Context	Memilih atribut dalam data transaksi yang akan digunakan dalam proses asosiasi	
Precondition	Data masukan valid	
Successful End Condition	Sistem berhasil memilih atribut yang akan digunakan dalam proses asosiasi	
Failed End Condition	Sistem tidak berhasil memilih atribut yang akan digunakan dalam proses asosiasi	
Actors	Manager	
Triger	User mengklik tombol *Pilih Atribut*	
Main Flow	Step	Action
	1	User melakukan *request* pemilihan atribut
	2	Sistem melakukan proses pemilihan terhadap kolom data transaksi
	3	Sistem menampilkan pemilihan atribut
Extension	Step	Branching Action
	-	-

4) *Use Case Scenario Pembersihan Data*

Scenario use case Pembersihan Data menggambarkan langkah-langkah melakukan pembersihan terhadap data transaksi penjualan yang hanya memiliki satu pembelian dalam satu transaksi. Scenario use case *Diagram Activity* data dapat dilihat pada table 3.29.

Tabel 3. 29 Use Case Scenario *Pembersihan Data*

Use case Name	***Pembersihan Data***
Goal In Context	Membersihkan data transaksi yang hanya memiliki satu pembelian dalam satu transaksi

Precondition	Data sudah dilakukan pemilihan atribut	
Successful End Condition	Sistem berhasil membersihkan data pada transaksi yang hanya terdapat satu pembelian	
Failed End Condition	Sistem tidak berhasil membersihkan data data pada transaksi yang hanya terdapat satu pembelian	
Actors	Manager	
Triger	Sudah melewati proses *import data*	
Main Flow	Step	Action
	1	User melakukan *request Diagram Activity*
	2	Sistem melakukan proses *Diagram Activity*
	3	Sistem menampilkan hasil proses *Diagram Activity*
Extension	Step	Branching Action
	-	-

5) *Use Case Scenario*Frekuensi

*Scenario use case*hitung frekuensi menggambarkan langkah-langkah melakukan perhitungan frekuensi terhadap *Item* yang terdapat pada data transaksi. Scenario use case data hitung frekuensi dapat dilihat pada tabel 3.29.

Tabel 3. 30 Use Case ScenarioFrekuensi

Use case Name	**Hitung Frekuensi**	
Goal In Context	Menghitung jumlah kemunculan tiap *Item* dalam data transaksi	
Precondition	Posisi data sudah dilakukan *Diagram Activity*	
Successful End Condition	Sistem berhasil menampilkan frekuensi tiap *Item*	
Failed End Condition	Sistem tidak berhasil melakukan perhitungan frekuensi*Item*	
Actors	Manager	
Triger	User menekan tombol Frekuensi	
Main Flow	Step	Action
	1	User melakukan *request*frekeunsi
	2	Sistem mengambil data hasil *Diagram Activity*
	3	Sistem melakukan proses penghitungan frekuensi tiap *Item*
	4	Sistem menampilkan *Item* beserta frekuensi kemunculannya
Extension	Step	Branching Action
	-	-

6) Use Case Scenario *priority*

Scenario use case *priority* menggambarkan langkah-langkah melakukan pengurutan data berdasarkan *priority*. Scenario use case *priority* dapat dilihat pada tabel 3.30.

Tabel 3. 31 Use Case Scenario *priority*

Use case Name	**Pengurutan Data**	
Goal In Context	Data terurut berdasarkan *priority*	
Precondition	Data sudah dihitung frekuensinya	
Successful End Condition	Sistem berhasil melakukan pengurutan berdasarkan *priority*	
Failed End Condition	Sistem tidak berhasil melakukan pengurutan data berdasarkan *priority*	
Actors	Manager	
Triger	User menekan tombol *priority*	
Main Flow	Step	Action
	1	User melakukan *request priority*
	2	Sistem melakukan proses pengurutan pada setiap transaksi berdasarkan prioritasnya
	3	Sistem menampilkan data yang sudah terurut
Extension	Step	Branching Action
	-	-

7) *Use Case ScenarioAsosiasi*

Skenario *use case*proses *assosiasi* menggambarkan langkah-langkah aksi *user* untuk melakukan proses *frequentItemset*, *rule*, dan melihat hasil asosiasi. Skenario *use case* proses asosiasi dapat dilihat pada table 3.31.

Tabel 3. 32 Use Case Scenario Assosiasi

Use case Name	**Proses Assosiasi**	
Goal In Context	Sistem mendapatkan *rule* lalu menampilkan *rule* tersebut menjadi sebuah informasi	
Precondition	telah melakukan proses *preprocessing*	
Successful End Condition	Sistem berhasil mendapatkan *rule* lalu menampilkan *rule* tersebut menjadi sebuah informasi	
Failed End Condition	Sistem tidak berhasil mendapatkan *rule*	
Actors	Manager	
Triger	*User* memasukan nilai *minimum support* dan *minimum confidence* yang valid	
Main Flow	Step	Action
	1	*User* memasukan nilai minimum *support*
	2	*User* memasukan nilai minimum *confidence*
	3	Sistem melakukan *generaterule*

	4	Sistem melakukan seleksi *rule* sesuai nilai minimum *confidence*
	5	Sistem mendapatkan *rule* yang sesuai dengan minimum *confidence*
	6	Sistem menampilkan hasil seleksi *rule*
Extension	Step	Branching Action
	1.1	Sistem menampilkan notifikasi nilai *minimum support* yang diinputkan tidak sesuai
	1.2	Sistem menampilkan notifikasi nilai minimum *confidence* yang diinputkan tidak sesuai
	1.3	Sistem tidak mendapatkan *rule* karena nilai minimum *confidence* yang diinputkan oleh user terlalu tinggi

8) Skenario *Use case* Menampilkan *Rule*

Skenario *use case* menampilkan *rule* menggambarkan langkah-langkah aksi *user* untuk melakukan proses pembuatan *rule*. Skenario *use case rule*dapat dilihat pada table 3.32.

Tabel 3. 33 Use Case Menampilkan Rule

Use case Name	***Rule***	
Goal In Context	Sistem melakukan pembentukan *rule*	
Precondition	Sudah melewati proses *input nilai minimum support dan confidence*	
Successful End Condition	Sistem berhasil melakukan pembentukan *rule*	
Failed End Condition	Sistem tidak berhasil melakukan pembentukan *rule*	
Actors	Pemilik	
Triger	*User* menginputkan nilai minimum *confidence*	
Included Cases	*minimum confidence*	
Main Flow	Step	Action
	1	*User* menginputkan nilai minimum *confidence*
	2	Sistem mendapatkan nilai minimum *confidence*
	3	Sistem melakukan *generate rule*
	4	Sistem memilah *rule* sesuai dengan minimum *confidence*
	5	Sistem mendapatkan *rule* yang sesuai dengan minimum *confidence*
Extension	Step	Branching Action
	1.1	Sistem menampilkan pesan nilai minimum *confidence* yang diinputkan tidak sesuai
	1.2	Sistem gagal mendapatkan *rule* karena nilai *confidence* yang diinputkan oleh *user* terlalu tinggi

9) Skenario *Use Case* Pembentukan Paket

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

Tabel 3. 34 Skenario *Use Case* Pembentukan Paket

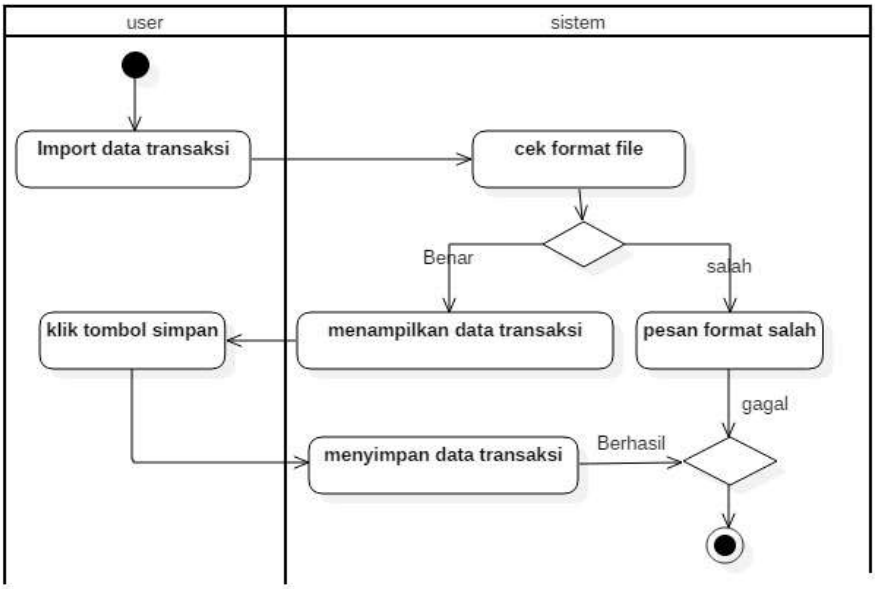
Use case Name	***Rule***	
Goal In Context	Sistem mendapatkan *rule* yang kemudian akan ditampilkan menjadi sebuah informasi	
Precondition	Data sudah melalui proses *Asosiasi*	
Successful End Condition	Sistem berhasil mendapatkan *rule* kemudian menampilkan hasil rule	
Failed End Condition	Sistem gagal mendapatkan *rule* yang akan ditampilkan dalam bentuk informasi	
Actors	Pemilik	
Triger	*User* menginputkan nilai parameter berupa nilai minimal harga dan maksimum harga	
Included Cases	Input Parameter harga, Lihat Pembentukan Paket	
Main Flow	Step	Action
	1	*User* menginputkan minimum *Harga*
	2	*User* memasukan nilai maksimum harga
	3	Sistem menghitung minimum harga dan maksimum harga
	4	Sistem menghapus data yang kurang dari nilai minimum harga dan maksimum harga
	5	Sistem menampilkan data paket yang memenuhi batasan harga
Extension	Step	Branching Action
	1.1	Sistem menampilkan notifikasi nilai minimum harga yang dimasukan tidak sesuai
	1.2	Sistem menampilkan notifikasi nilai maksimum harga yang dimasukan tidak sesuai
	1.3	Sistem tidak mendapatkan *rule* karena nilai Minimum harga dan maksimum harga tidak sesuai.

3. *Activity Diagram*

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

1) *Activity Diagram Import Data*

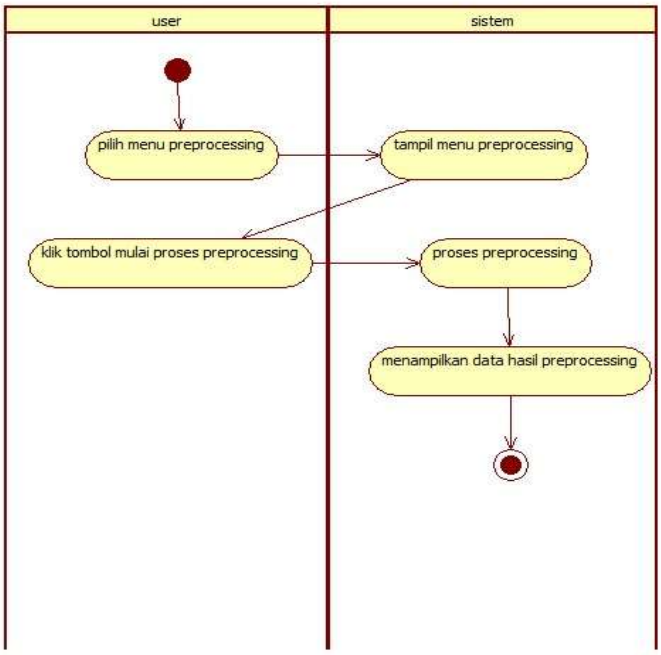
Activity Diagram import data menjelaskan aktivitas aktor dalam melakukan proses import data. *Activity Diagram*import data dapat dilihat pada Gambar 3.70



Gambar 3. 70 *Activity Diagram Import Data*

2) *Activity Diagram* Prepocessing

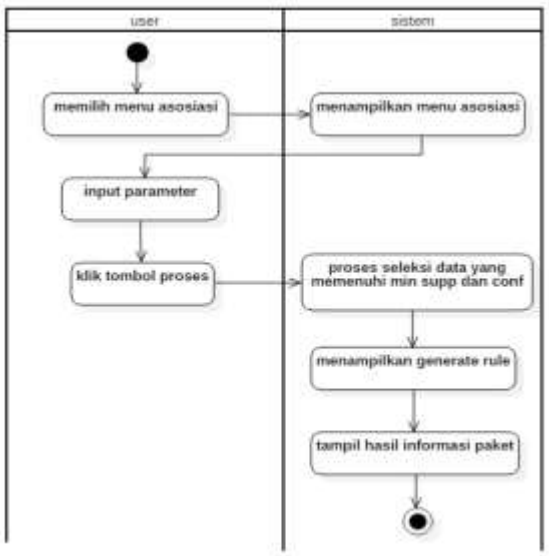
Activity Diagram preprocessing menjelaskan aktivitas sistem dalam melakukan proses pengolahan data. *Diagram Aktivitas* preprocessing dapat dilihat pada gambar 3.71



Gambar 3. 71 Activity Diagram Processing

6) *Activity Diagram* Asosiasi

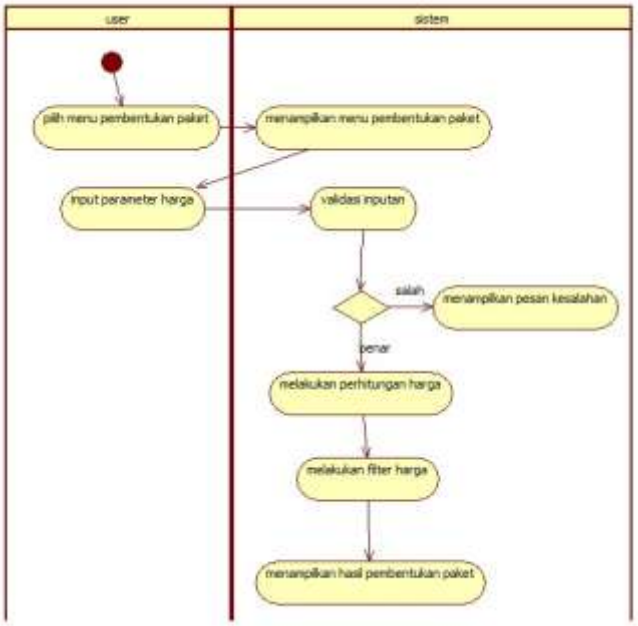
Activity Diagram proses *asosiasi* menjelaskan aktivitas *user* dan sistem dalam melakukan proses pencarian *frequent Itemset , rule* dan melihat hasil asosiasi. *Diagram Aktivitas proses asosiasi* dapat dilihat pada gambar 3.76



Gambar 3. 72 *Activity Diagram* Asosiasi

7) *Activity Diagram* Pembentukan Paket

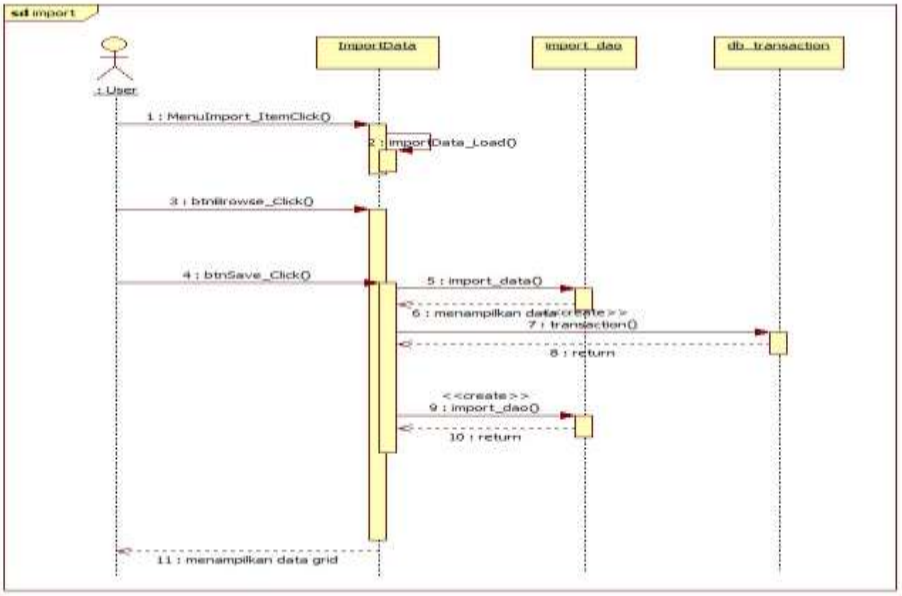
Activity diagram pembentukan paket menjelaskan aktivitas *user* dan sistem dalam menentukan minimal harga dan maksimal harga sesuai dengan yang dimasukkan oleh user. *Activity diagram* lihat hasil pembentukan paket dapat dilihat pada gambar 3.73



Gambar 3. 73 *Activity Diagram* Pembentukan Paket

4. *Sequence Diagram*
    1. *Sequence Diagram Import Data*

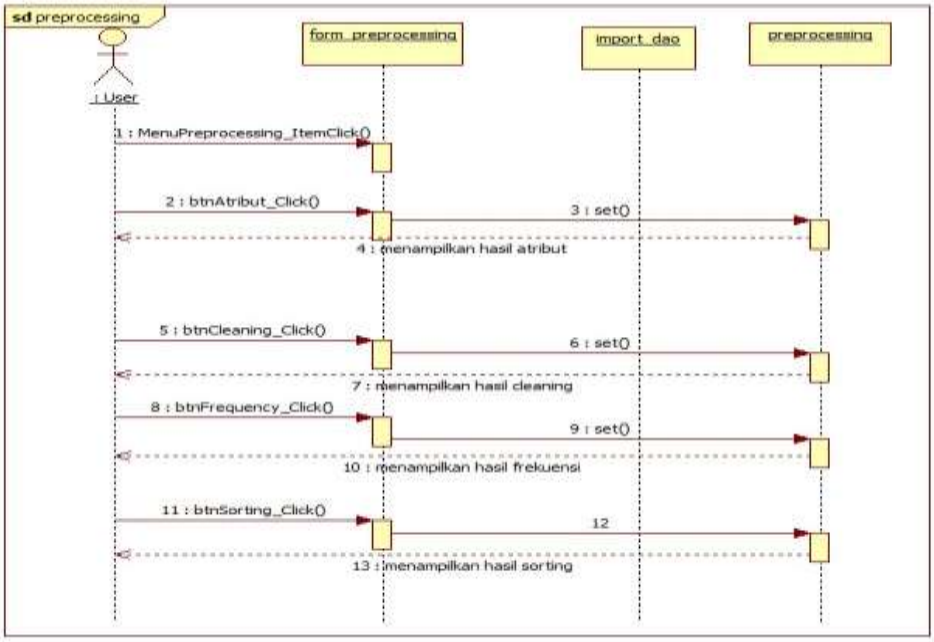
        Sequence Diagram Import Data dapat dilihat pada gambar 3.145



Gambar 3. 74 *Sequence Diagram Import Data*

2. *Sequence Diagram Preprocessing*

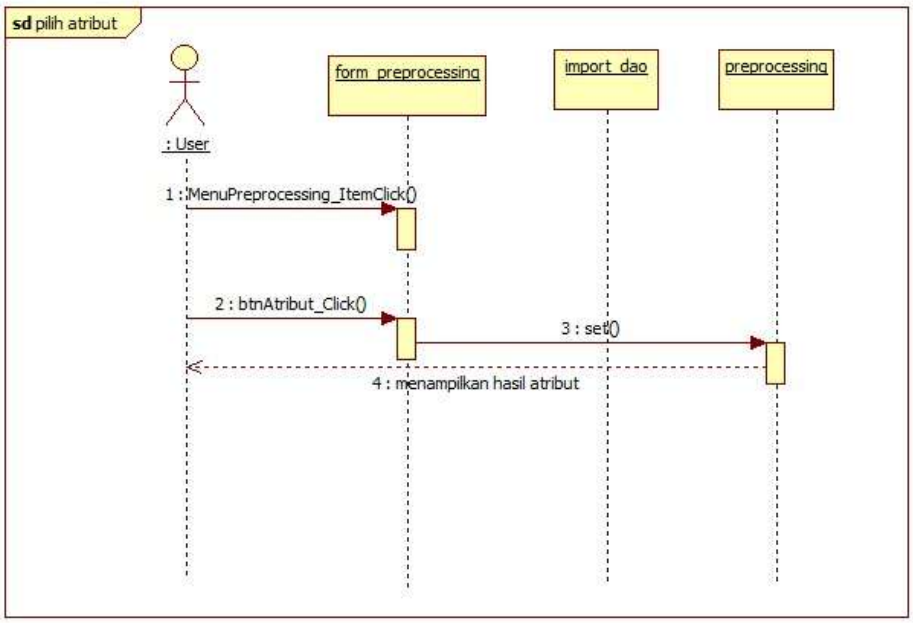
Sequence Diagram Preprocessing dapat dilihat pda gambar 3.146



Gambar 3. 75 *Sequence Diagram Preprocessing*

a. *Sequence Diagram* Pemilihan Data

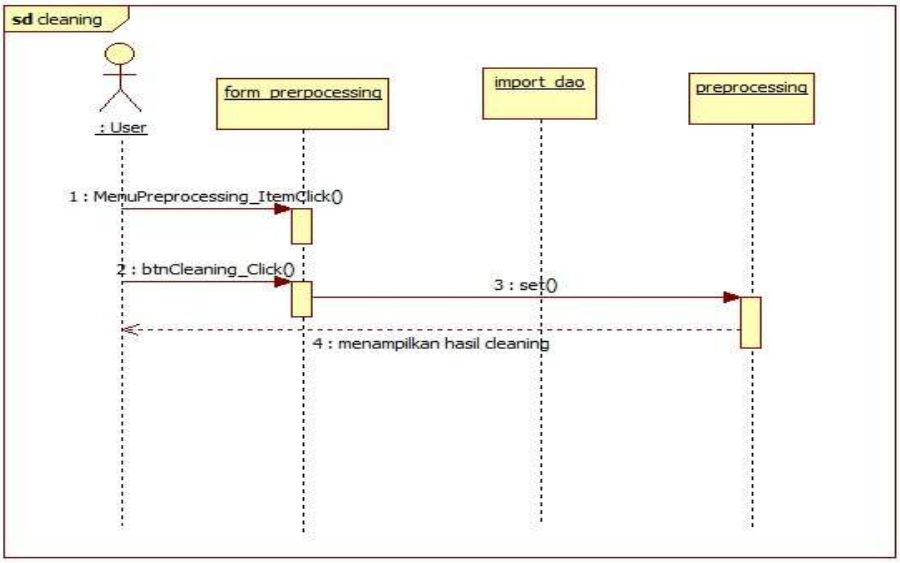
Sequence Diagram Pemilihan data dapat dilihat pada gambar 3.147



Gambar 3. 76 *Sequence Diagram* Pemilihan Data

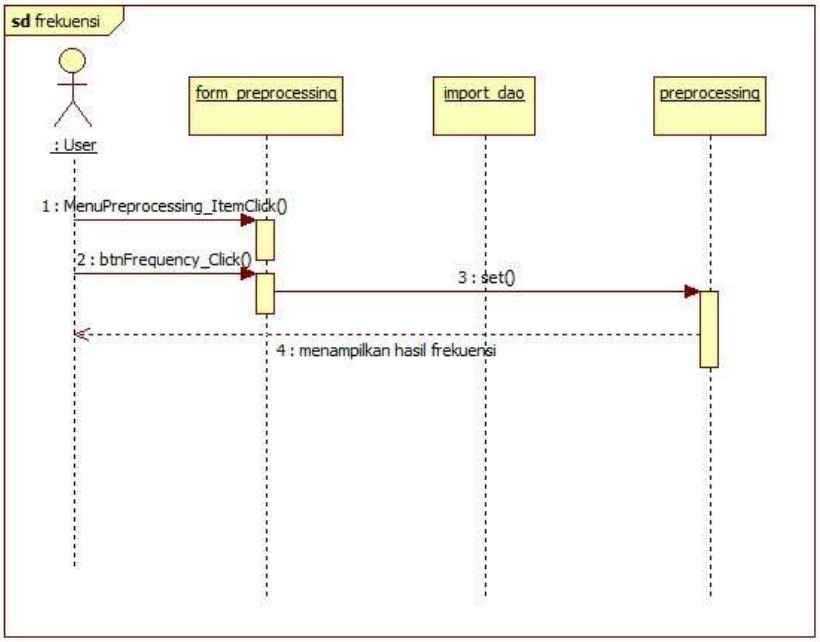
b. *Sequence Diagram Cleaning*

Sequence Diagram Cleaning dapat dilihat pada gambar 3.148



c. *Sequence Diagram* Proses Hitung Frekuensi

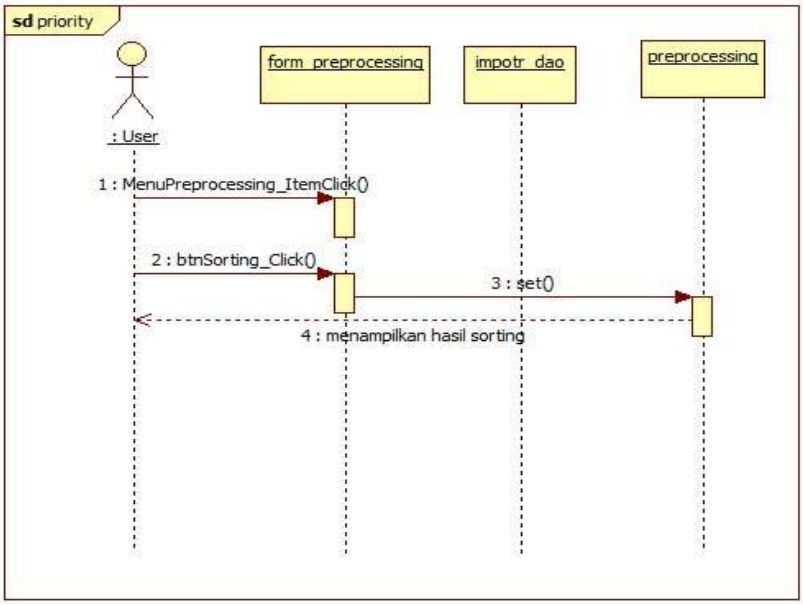
Sequence Diagram Proses Hitung Frekuensi dapat dilihat pada



Gambar 3. 78 *Sequence Diagram* Proses Hitung Frekuensi

d. *Sequence Diagram Priority*

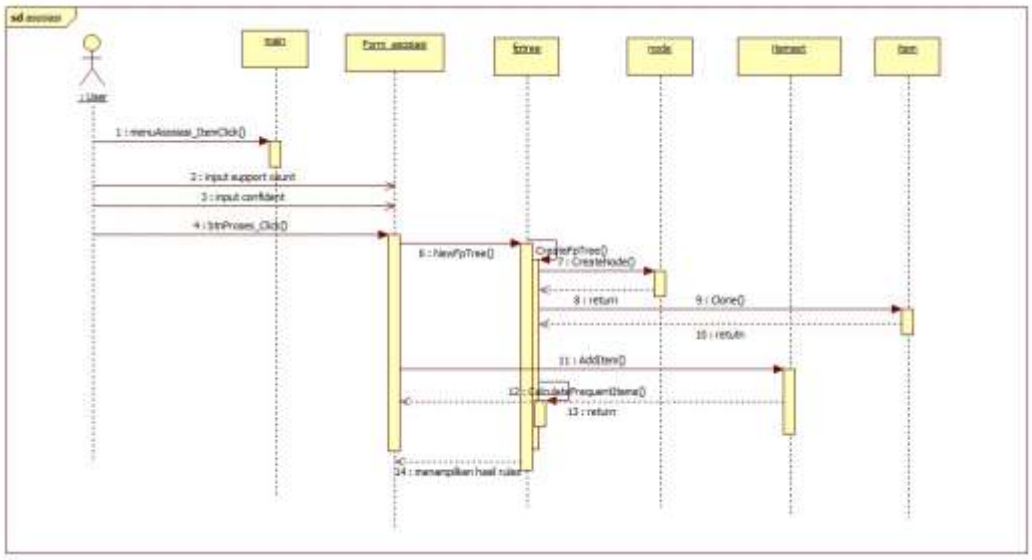
Sequence Diagram Priority dapat dilihat pada gambar 3.150



Gambar 3. 79 *Sequence Diagram Priority*

1. *Sequence Diagram* Asosiasi

   Sequence Diagram Asosiasi dapat dilihat pada gambar 3.151

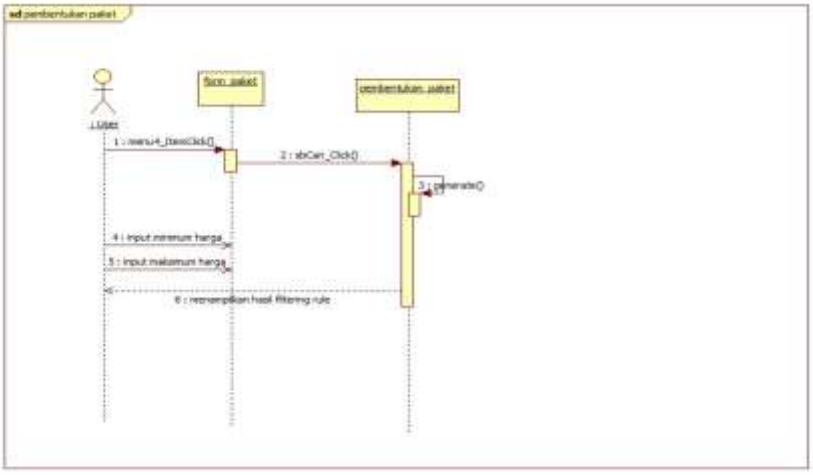


Gambar 3. 80 *Sequence Diagram* Asosisasi

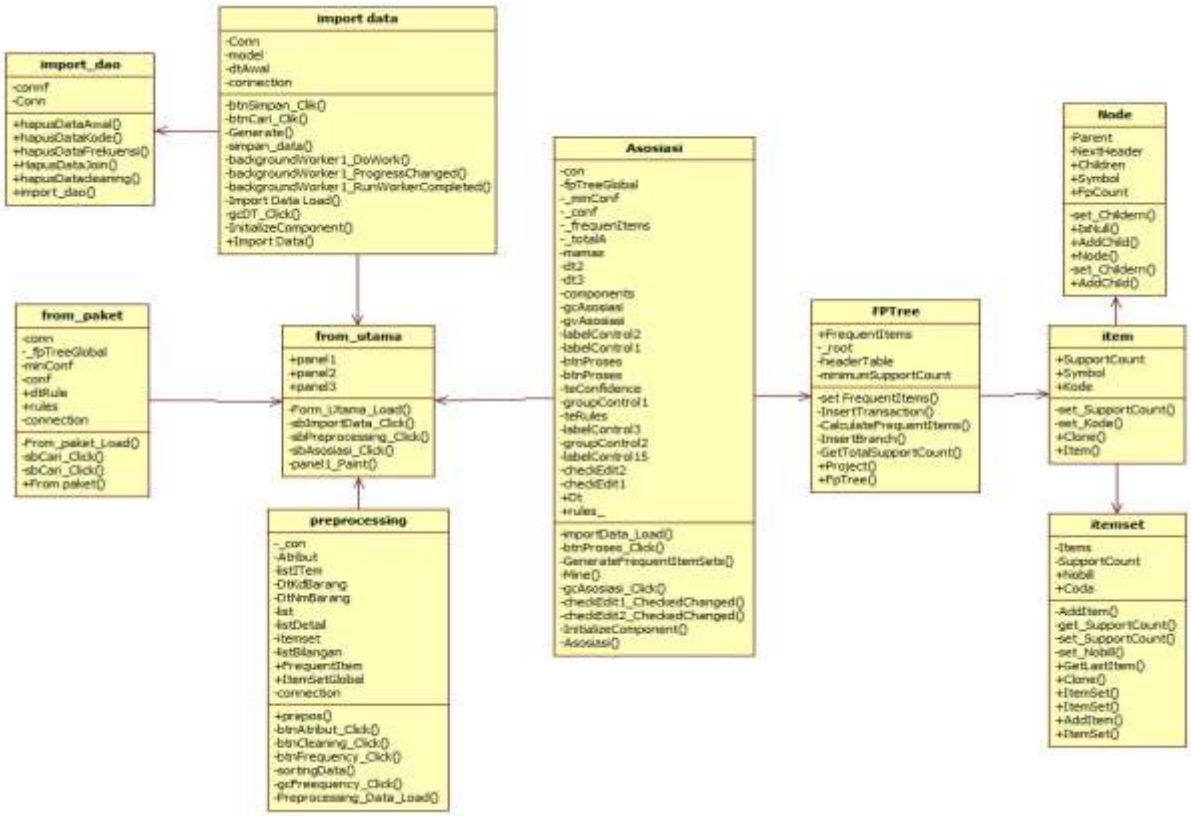
a. *Sequence Diagram* Pembentukan Paket

   Sequence Diagram Pembentukan Paket dapat dilihat pada gambar

2. *Class Diagram Class Diagram* dapat dilihat pada gambar dibawah ini



Gambar 3. 81 *Sequence Diagram* Pembentukan Paket



Gambar 3. 82 *Class Diagram*

3.2 Perancangan Sistem

Proses perancangan yang akan dilakukan pada tahapan ini adalah perancangan sistem basis data, perancangan stuktur menu, perancangan antar muka, perancangan pesan, jaringan semantik.

3.2.1 Perancangan *Class*

Berikut ini adalah perancangan *Class* yang digunakan pada sistem yang akan dibangun :

Tabel 3. 35 Struktur *Form_Utama*

Class : Form_Utama		
Nama	Jenis	Type
panel1	*Atribut*	*Public*
panel2	*Atribut*	*Public*
panel3	*Atribut*	*Public*
Form_Utama_Load	*Method*	*Private*
sbImportData_Click	*Method*	*Private*
sbPreprocessing_Click	*Method*	*Private*
sbAsosiasi_Click	*Method*	*Private*
panel1_Paint	*Method*	*Private*

Tabel 3. 36 Struktur *Class import_Data*

Class : Import_data		
Nama	Jenis	Type
Conn	*Atribut*	*Private*
Model	*Atribut*	*Private*
dtAwal	*Atribut*	*Private*
Connection	*Atribut*	*Private*
btnSimpan_Click	*Method*	*Private*
btnCari_Click	*Method*	*Private*
Generate	*Method*	*Private*
simpan_data	*Method*	*Private*
backgroundWorker1_DoWork	*Method*	*Private*
backgroundWorker1_ProgressChanged	*Method*	*Private*
backgroundWorker1_RunWorkerCompleted	*Method*	*Private*
Import_Data_Load	*Method*	*Private*
gcDT_Click	*Method*	*Private*
InitializeComponent	*Method*	*Private*
Import_Data	*Method*	*Public*

Tabel 3. 37 Struktur *Class Import_dao*

Class : Import_dao		
Nama	Jenis	Type
connf	*Atribut*	*Private*
Conn	*Atribut*	*Private*
hapusDataAwal	*Method*	*Public*
hapusDataKode	*Method*	*Public*
hapusDataFrekuensi	*Method*	*Public*
hapusDataJoin	*Method*	*Public*
hapusDatacleaning	*Method*	*Public*
import_dao	*Method*	*Public*

Tabel 3. 38 Struktur *Class Prerpocessing Data*

Class : preprocessing		
Nama	Jenis	Type
con	*Atribut*	*Private*
atribut	*Atribut*	*Private*
listITem	*Atribut*	*Private*

DtKdBarang	*Atribut*	*Private*
DtNmBarang	*Atribut*	*Private*
list	*Atribut*	*Private*
listDetail	*Atribut*	*Private*
itemSet	*Atribut*	*Private*
listBilangan	*Atribut*	*Private*
FrequentItems	*Atribut*	*Public*
ItemSetGlobal	*Atribut*	*Public*
connection	*Atribut*	*Private*
prepos	*Method*	*Public*
btnAtribut_Click	*Method*	*Private*
btnCleaning_Click	*Method*	*Private*
btnFrequency_Click	*Method*	*Private*
sortingData	*Method*	*Private*
gcFreequency_Click	*Method*	*Private*
Preprocessing_Data_Load	*Method*	*Private*

Tabel 3. 39 Struktur *Class Asosiasi*

Class : asosiasi		
Nama	Jenis	Type
Con	*Atribut*	*Private*
_fpTreeGlobal	*Atribut*	*Private*
_minConf	*Atribut*	*Private*
_conf	*Atribut*	*Private*
_frequentItems	*Atribut*	*Private*
_itemSetGlobal	*Atribut*	*Private*
_totalAb	*Atribut*	*Private*
_totalA	*Atribut*	*Private*
_mamas	*Atribut*	*Private*
dt2	*Atribut*	*Private*
dt3	*Atribut*	*Private*
components	*Atribut*	*Private*
gcAsosiasi	*Atribut*	*Private*
gvAsosiasi	*Atribut*	*Private*
labelControl2	*Atribut*	*Private*
labelControl1	*Atribut*	*Private*
btnProses	*Atribut*	*Private*
teConfidence	*Atribut*	*Private*
teSupport	*Atribut*	*Private*
groupControl1	*Atribut*	*Private*

teRules	*Atribut*	*Private*
labelControl3	*Atribut*	*Private*
groupControl2	*Atribut*	*Private*
labelControl5	*Atribut*	*Private*
checkEdit2	*Atribut*	*Private*
checkEdit1	*Atribut*	*Private*
Dt	*Atribut*	*Public*
rules_	*Atribut*	*Public*
importData_Load	*Method*	*Private*
btnProses_Click	*Method*	*Private*
GenerateFrequentItemSets	*Method*	*Private*
Mine	*Method*	*Private*
gcAsosiasi_Click	*Method*	*Private*
checkEdit1_CheckedChanged	*Method*	*Private*
checkEdit2_CheckedChanged	*Method*	*Private*
InitializeComponent	*Method*	*Private*
Asosiasi	*Method*	*Private*

Tabel 3. 40 Struktur *Class Form_paket*

Class : form_paket		
Nama	Jenis	Type
_conn	*Atribut*	*Private*
_fpTreeGlobal	*Atribut*	*Private*
_minConf	*Atribut*	*Private*
_conf	*Atribut*	*Private*
dtRule	*Atribut*	*Public*
rules_	*Atribut*	*Public*
connection	*Atribut*	*Private*
Form_paket_Load	*Method*	*Private*
sbCari_Click	*Method*	*Private*
sbCari_Click	*Method*	*Private*
Form_paket	*Method*	*Public*

Tabel 3. 41 Struktur *Class FPTree*

Class : FPTree		
Nama	Jenis	Type
FrequentItems	*Atribut*	*Public*
_root	*Atribut*	*Private*
_headerTable	*Atribut*	*Private*
_minimumSupportCount	*Atribut*	*Private*
set_FrequentItems	*Method*	*Private*
InsertTransaction	*Method*	*Private*

CalculateFrequentItems	*Method*	*Private*
InsertBranch	*Method*	*Private*
GetTotalSupportCount	*Method*	*Private*
Project	*Method*	*Public*
FpTree	*Method*	*Public*

Tabel 3. 42 Struktur *Class Node*

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

Tabel 3. 43 Struktur *Class Item*

Class : Item		
Nama	Jenis	Type
SupportCount	*Atribut*	*Public*
Symbol	*Atribut*	*Public*
Kode	*Atribut*	*Public*
set_SupportCount	*Method*	*Private*
set_Kode	*Method*	*Private*
Clone	*Method*	*Public*
Item	*Method*	*Public*

Tabel 3. 44 Struktur *Class Itemset*

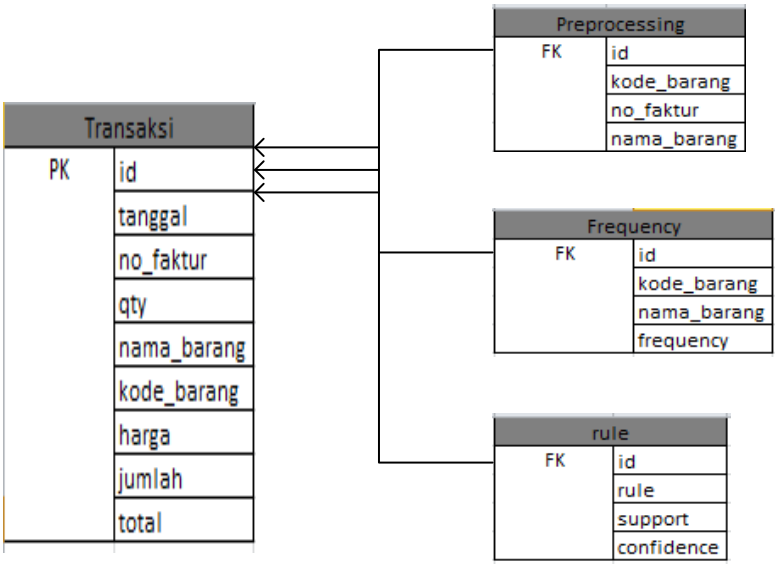
Class : Itemset		
Nama	Jenis	Type
Items	*Atribut*	*Private*
SupportCount	*Atribut*	*Private*
Nobill	*Atribut*	*Public*
Coda	*Atribut*	*Public*
AddItem	*Method*	*Private*
get_SupportCount	*Method*	*Private*
set_SupportCount	*Method*	*Private*
set_Nobill	*Method*	*Private*
GetLastItem	*Method*	*Public*
Clone	*Method*	*Public*
ItemSet	*Method*	*Public*
ItemSet	*Method*	*Public*
AddItem	*Method*	*Public*
ItemSet	*Method*	*Public*

3.2.2 Perancangan Basis data

Perancangan basis data terhadap aplikasi yang akan dibangun meliputi struktur tabel.

3.2.2.1 Skema Relasi

Skema relasi dari aplikasi yang akan dibangun dapat dilihat pada gambar 3. 152 dibawah ini.



Gambar 3. 83 Skema Relasi

3.2.2.2 Struktur Tabel

Struktur table pada aplikasi yang akan dibangun terdiri dari 4 tabel, rinciannya sebagai berikut

Tabel 3. 45 Struktur Tabel Data Transaksi

Nama Field	Tipe Data	Panjang	Keterangan
id	Int	11	PK
tanggal	Varchar	20	
no_faktur	Varchar	20	
qty	Int	11	
nama_barang	Varchar	50	

kode_barang	Varchar	20	
harga	Int	11	
jumlah	Int	11	
total	Float		

Tabel 3. 46 Tabel Frequency

Nama Field	Tipe Data	Panjang	Keterangan
id	Int	11	FK references Transaksi(Id)
kode_barang	Varchar	10	
nama_barang	Varchar	100	

Tabel 3. 47 Tabel Preprocessing

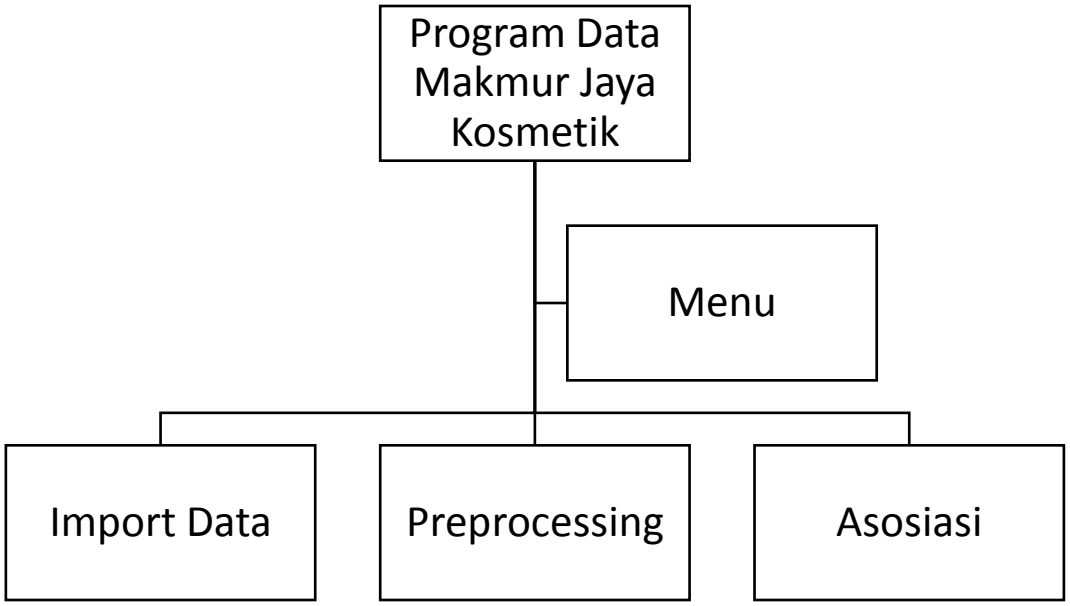
Nama Field	Tipe Data	Panjang	Keterangan
id	Int	11	FK references Transaksi(Id)
kode_barang	Varchar	10	
no_faktur	Varchar	50	
nama_barang	Varchar	100	

Tabel 3. 48 Tabel *Rule*

Nama Field	Tipe Data	Panjang	Keterangan
Rule	Text	-	FK references Transaksi(Id
Support	Int	11	
Confidence	Nt	11	

3.2.6 Perancangan Struktur Menu

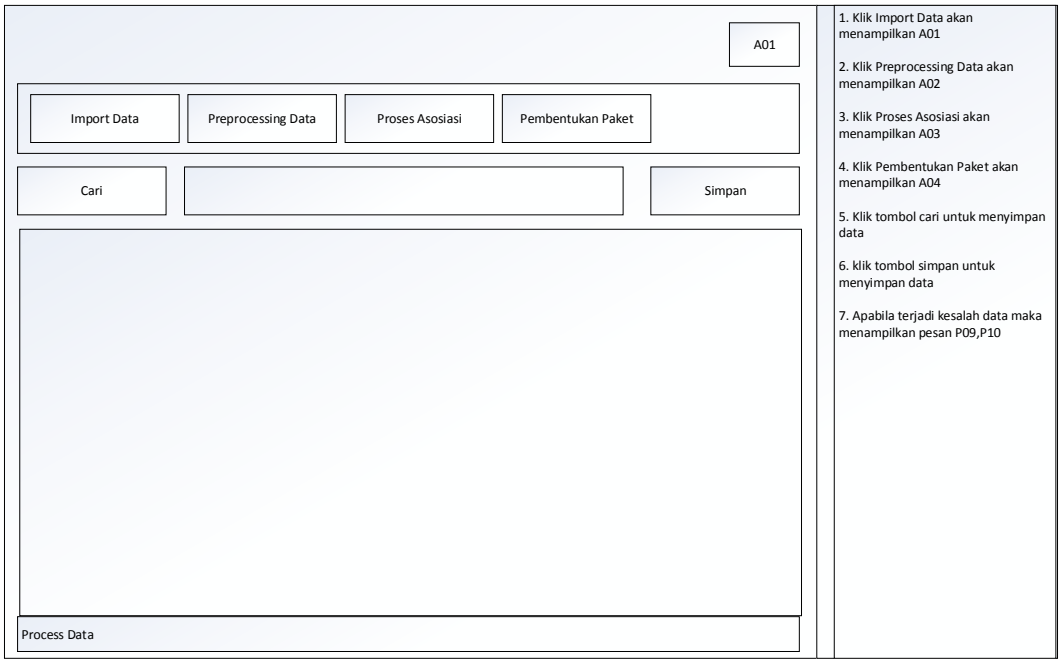
Dalam pembangunan perangkat lunak diperlukan struktur menu untuk memudahkan pengguna dalam menavigasi fungsional sistem. Adapun perancangan struktur menu untuk aplikasi yang akan dibangun dapat dilihat pada gambar berikut :



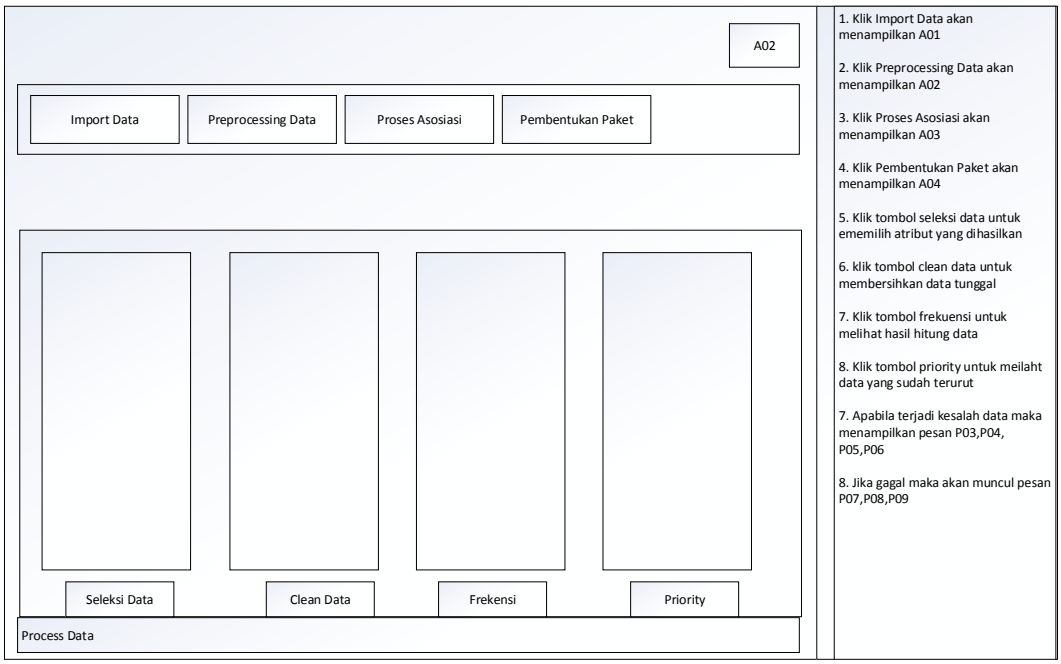
Gambar 3. 84 Perancangan Struktur Menu

3.2.7 Perancangan Antar Muka

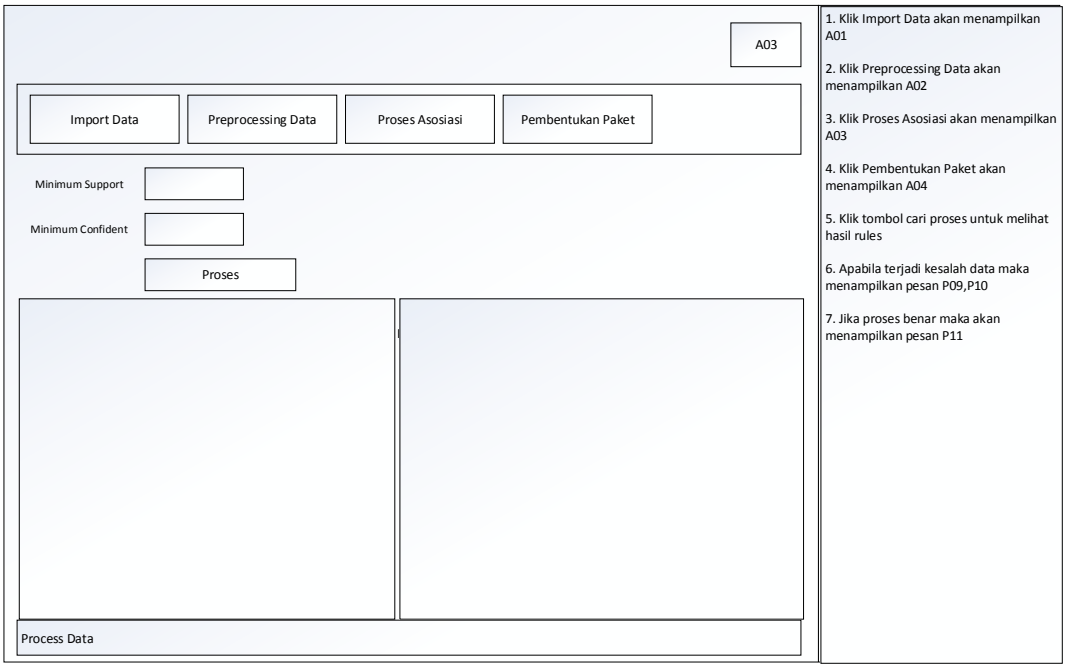
Perancangan antar muka berfungsi untuk memberi gambaran tentang pernagkat lunak yang akan dibangun, supaya lebih memudahkan dalam pembuatan perangkat lunak nantinya. Adapun perancangan antar muka untuk perangakt lunak yang akan dibangun akan dijelaskan berikut ini :



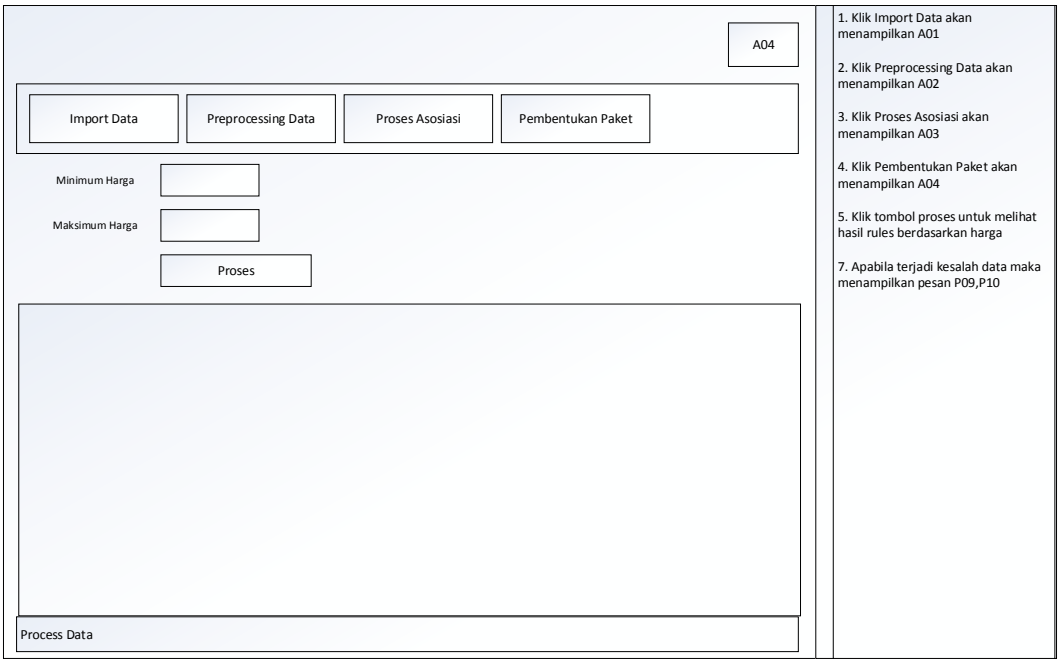
Gambar 3. 85 Perancangan Input Data



Gambar 3. 86 Perancangan Antar Muka Preprocessing



Gambar 3. 87 Perancangan Antar Muka Aturan Asosiasi



Gambar 3. 88 Perancangan Antar Muka Pembentukan Paket

3.2.5 Perancangan Pesan

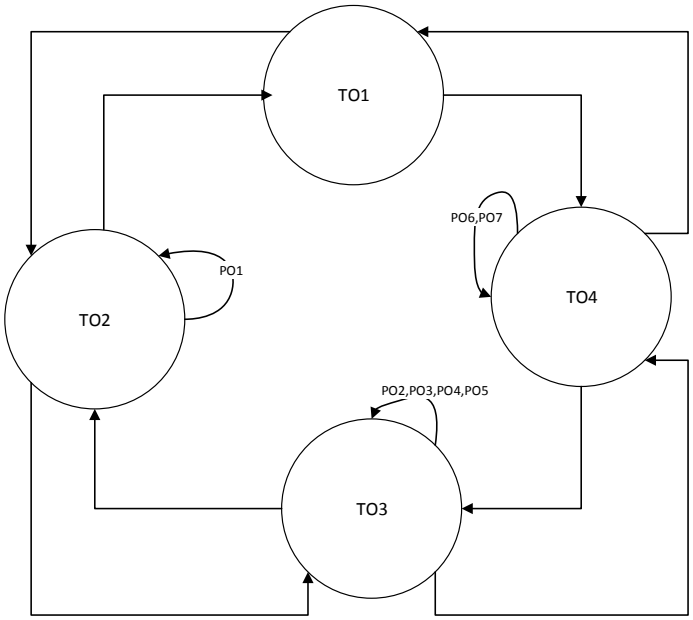
Adapun perancangan pesan dapat dilihat pada gambar dibawah ini :



Gambar 3. 89 Perancangan Pesan

3.2.8 Jaringan Semantik

Jaringan semantik yang menghubungkan antarmuka pada aplikasi yang akan dibangun dapat dilihat pada gambar berikut ini :



Gambar 3. 90 Jaringan Semantik