BAB 3
ANALISIS DAN PERANCANGAN

3.1.1 Analisis Sistem

Analisis sistem didefinisikan sebagai penguraian suatu sistem informasi yang utuh kedalam beberapa bagian dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan ataupun hambatan-hambatan yang terjadi untuk menghasilkan sebuah solusi guna melakukan perbaikan-perbaikan terhadap sistem yang sedang berjalan. Dalam analisa sistem ini meliputi beberapa bagian, yaitu :

1. Analisis Masalah
2. Analisis *Data Mining*
3. Analisis Spesifikasi Kebutuhan Perangkat Lunak
4. Analisis Kebutuhan Non-Fungsional
5. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

Berdasarkan hasil wawancara dengan pihak AIRPLANE SYSTM, dapat disimpulkan bahwa permasalahannya yakni:

1. Banyaknya art-art yang dihasilkan oleh AIRPLANE SYSTM pada setiap bulannya membuat bagian produksi kurang hati-hati dalam memilih art untuk diproduksi pada bulan berikutnya.
2. Menentukan art yang akan diproduksi berdasarkan *history* penjualan dibeberapa pemesanan oleh *space dealer* dan pihak AIRPLANE SYSTM sendiri yang beberapa bulan kebelakang mengalami penurunan karena kesalahan pemilihan art yang kurang tepat.

3.1.2 Analisis *Data Mining*

Berdasarkan tahapan-tahapan yang terdapat dalam CRIPS-DM, maka dalam penelitian *data mining* di AIRPLANE SYSTM ini terdapat beberapa tahapan sebagai berikut:

3.1.2.1 Pemahaman Bisnis

Tahapan prmahaman bisnis merupakan tahapan pertama yang dilakukan dalam kerangka kerja CRISP-DM. dalam tahapan bisnis ini terdapat beberapa tahapan lainnya, yaitu :

1. Identifikasi Tujuan Bisnis

   Dalam proses bisnisnya AIRPLANE SYSTM memasarkan produk-produknya langsung kepada konsumen dan kepada *space dealer* (sebutan agen retail). Untuk membuat strategi bisnis perusahaan, pihak *marketing* harus menganalisa data penjualan. Tujuan dari menganalisa data penjualan yakni untuk mengetahui kecenderungan pemilihan produk yang sering dibeli dengan cara mengetahui pola pemesanan *space dealer* agar produk yang diproduksi tepat sasaran.

2. Penentuan Sasaran *Data Mining*

   Tujuan *data mining* untuk perusahaan adalah menggali pengetahuan (*discovering knowledge*) tentang pola (*pattern*) produk-produk apa saja yang sering dibeli *space dealer* secara bersaman dalam sebuah transaksi penjualan sehingga diketahui produk mana saja yang sering dipesan *space dealer* secara bersamaan agar pihak perusahaan mendapatkan gambaran dalam menentukan produk mana saja yang akan diproduksi.

3.1.2.2 Pemahaman Data

Tahapan pemahaman data merupakan tahapan kedua yang dilakukan setelah tahapan bisnis dalam kerangka kerja CRISP-DM. dalam tahap pemahaman data ini terdapat beberapa langkah diantaranya :

a. Pengumpulan data awal

   Data yang digunakan dalam penelitian ini yakni data penjualan produk berdasarkan pesanan dari beberapa *space dealer* pada bulan Januari sampai dengan bulan Desember 2014. Data yang terlibat dalam data penjualan produk ini adalah data space *dealer outlet,* data yang digunakan untuk proses analisis berupa file Excel dengan format *.csv atau *.xlsx.

   Data yang digunakan untuk proses perhitungan secara manual dengan mengambil sebanyak 150 data untuk dilakukan pengolahan ke tahap

selanjutnya. Adapun detail informasi mengenai data transaksi yang digunakan dapat dilihat dalam table 3.1 dibawah ini.

Tabel 3.1 Struktur Data Penjualan

DOKUMEN	KETERANGAN		
Detail data penjualan	Deskripsi :	Data ini berisi mengenai pemesanan barang dari beberapa retail pada satu bulan di AIRPLANE SYSTM	
	Fungsi :	Untuk mengetahui produk mana saja yang paling banyak dipesan pada bulan tersebut	
	Format :	Microsoft Excel (.xlsx)	
	Atribut :	No NOTA	Kode dari produk yang dipesan
		Tanggal	Waktu pemesanan Barang
		Brand	Tempat produksi baju dibuat
		Sex	Jenis produk yang di pesan
		Art	Nama barang yang dipesan
		Colour	Warna dari produk
		Size	Ukuran dari produk
		Qty	Jumlah pemesanan
		Price	Harga produk
		Amount	Jumlah harga produk yang dipesan

Berikut adalah contoh data penjualan pada bulan Desember yang didapat dari pesanan dari *space dealer* yang ada di AIRPLANE SYSTM yang akan digunakan untuk proses perhitungan metode *association rule* dengan algoritma FP-Growth, dimana atribut-atribut yang terdapat dalam data penjualan tersebut antara lain no nota, tanggal, *brand*, *sex*, *art*, *colour*, *size*, *qty*, *price*, *amount*. Untuk lebih jelasnya, dapat dilihat pada lampiran tabel D-1 di lembar lampiran.

3.1.2.3 *Preprocessing* Data

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

1. Ekstraksi data

   Ekstraksi data dapat diartikan sebagai proses pengambilan data dari sumber data dalam rangka untuk melanjutkan proses pengolahan data ke tingkat

selanjutnya ataupun untuk menyimpan data hasil ekstrak tersebut. Dalam penelitian ini, data yang berasal dari *flat file* berupa *microsoft excel* (.xlsx) diekstrak, kemudian disimpan ke dalam sebuah database agar memudahkan dalam proses pengolahan data.

2. Pembersihan data

   Proses menghilangkan *noise* dan menghilangkan data tidak relevan atau inkonsisten disebut dengan pembersihan data. Dalam penelitian ini data yang dihilangkan yaitu data yang pada kadang-kadang diproduksi seperti *wallet, jacket, pants, belt, sweater.* Berikut adalah hasil proses pembersihan data pada tabel D-2 di lembar lampiran.

3. Pemilihan atribut

   Pemilihan atribut adalah proses pemilihan atribut data yang akan digunakan, sehingga data tersebut dapat diproses sesuai dengan kebutuhan *data mining*. Dalam penelitian ini, atribut yang akan digunakan yakni no nota dan *Art*, karna penelitian ditujukan pada pola mengenai art, dapat dilihat pada lampiran tabel D-3 di lembar lampiran.

4. Setelah semua proses telah berhasil dilalui dan data transaksi telah sesuai dengan kebutuhan yang diperlukan dalam proses data mining, maka data penjualan tersebut sudah dapat digunakan untuk diproses dalam sistem *data mining* seperti terlihat dalam tabel 3.2.

Tabel 3.2 Hasil Penggabungan *Item* Per Nota

TID	NO NOTA	ART
1	AP001	Scrips,Rplnirish,Perspective
2	AP002	Ninetyeight,Manchester,Manfield,Girl,Floyd
3	AP003	Diminishing,Diamond,Culture,Clover,Cityelit
4	AP004	Asmilitary,Asdessert,As16
5	AP006	Blended,Machine,Phoenix,Manchester
6	AP007	Floyd,Lion
7	AP008	Tonic,Clover,Diminishingart,Handsline
8	AP009	Pure,Manfold,Cityelit,Rplntloseries,Cake

TID	NO NOTA	ART
9	AP010	Diamond,Manfield,Girl
10	AP011	Pure,Thundertype,Tloseries,Manfold
11	AP012	Manchester,Triangle,Scrips,Rplnirish,Systm,Manfield
12	AP013	Girl,Floyd,Diminishing,Diamond,Cityelit
13	AP014	Culture,Asmilitary,Asdessert,As16,Versty
14	AP015	Girl,Ninetyeight
15	AP016	Scrips,Diamond
16	AP017	Diminishing,Manfield,As16
17	AP018	Floyd,Versty,Perspective
18	AP019	Clover,Cityelit
19	AP020	Thundertype,Cake,Tloseries,Jerseytype,Handlines
20	AP021	Weed,Rplntloseries,Fireice,Tloarmy
21	AP022	Perspective,Manfold,Diminishing,Clover,As16
22	AP023	Clover,Culture
23	AP024	Diamond,Girl,Ninety,Manfield,Perspective,Scrips
24	AP026	Tloseries,Manmap,Thundertype,Pure,Jerseytype,Leafraglan
25	AP027	Weed,Fireandice,Stillhard,Tloarmy,Rplntloseries,Diamond
26	AP028	Scrips,Rplnirish,Perspective,Versty,Systm,Diminishing,As16
27	AP029	Jerseytype,As16,Thundertype

3.1.2.4 Pemodelan

Pemodelan merupakan tahap pembuatan model yang akan digunakan dalam proses *data mining*. Tahap ini dilakukan dengan menggunakan metode *association rule* yang akan memasuki ke dalam dua tahap yaitu :

a. *Frequent Itemset*

b. *Generate Rule*

Dalam tahapan *frequent itemset* akan dimasukan ke dalam model algoritma. Algoritma yang digunakan dalam penelitian ini adalah algoritma FP-Growth. Algoritma ini menggunakan pengetahuan mengenai *frequent itemset*

yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Dalam kasus ini sebelum melanjutkan ke proses selanjutnnya, untuk mempermudah pembentukan *frequent itemset* maka dibuatlah inisialisasi terhadap data yang digunakan. Berikut adalah hasil inisialisasi pada tabel 3.3 di bawah ini:

Tabel 3.3 Hasil Inisialisasi

ART	Inisial
Scrips	SC
Rplnirish	PH
Perspective	PE
Ninetyeight	NE
Manchester	MC
Manfield	MF
Girl	GI
Floyd	FD
Diminishing	DMG
Diamond	DMD
Culture	CT
Clover	CL
Cityelit	CE
Asmilitary	AM
Asdessert	AD
As16	AS
Blended	BLD
Machine	MCH
Phoenix	PX
Lion	LN
Pure	PU
Thundertype	TT

ART	Inisial
Tloseries	TS
Manfold	MFD
Triangel	TRI
Systm	SYS
Versty	VES
Tonic	TON
Diminishingart	DA
Handsline	HL
Rplntloseries	RTS
Cake	CK
Jerseytype	JT
Weeds	WE
Fireice	FI
Tloarmy	TA
Manmap	MM
Ninety	NIN
Fireandice	FAI
Stillhard	SH
Leafraglan	LR

Algoritma FP-Growth dibagi menjadi tiga langkah utama, namun terlebih dahulu dilakukan pembentukan pohon dengan menggunakan algoritma FP-Tree. Langkah-langkah algoritma FP-Growth yaitu:

1. Pembentukan pohon dengan FP-Tree

   FP-Tree merupakan struktur penyimpanan data yang dimapatkan. FP-Tree dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-Tree. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data FP-Tree semakin efektif. Kelebihan dari FP-Tree adalah hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien.

2. Langkah pertama adalah menentukan *minimum support* dari data yang digunakan yakni ada pada tabel 3.2 yang mana hasil dari penyederhanaan dalam lampiran tabel D-3. Dalam tahap yang selanjutnya nilai *minimum support* = 3 yang diambil dari hasil perhitungan rata-rata 1 *itemset* di bawah :

$$\frac{Jumlah\ data\ yang\ digunakan}{nama\ barang} = \frac{111}{42} = 2{,}6 \approx 3$$

   Dari hasil penyederhanaan data yang dilakukan pada setiap item yang ditunjukan pada tabel 3.4.

Tabel 3.4 Tabel yang telah disederhanakan

TID	NO NOTA	TRANSAKSI
1	AP001	SC, PH, PE
2	AP002	NE, MC, MF, GI, FD
3	AP003	DMG, DMD, CT, CL, CE
4	AP004	AM, AD, AS
5	AP006	BLD, MCH, PX, MC
6	AP007	FD, LN
7	AP008	TON, CL, DA, HL
8	AP009	PU, MFD, CE, RTS, CK

TID	NO NOTA	TRANSAKSI
9	AP010	DMD, MF, GI
10	AP011	PU, TT, TS, MFD
11	AP012	MC, TRI, SC, PH, SYS, MF
12	AP013	GI, FD, DMG, DMD, CE
13	AP014	CT, AM, AD, AS, VES
14	AP015	GI, NE
15	AP016	SC, DMD
16	AP017	DMG, MF, AS
17	AP018	FD, VES, PE
18	AP019	CL, CE
19	AP020	TT, CK, TS, JT, HL
20	AP021	WE, RTS, FI, TA
21	AP022	PE, MFD, DMG, CL, AS
22	AP023	CE, CL, CT
23	AP024	DMD, GI, NIN, MF, PE, SC
24	AP026	TS, MM, TT, PU, JT, LR
25	AP027	WE, FAI, SH, TA, RTS, DMD
26	AP028	SC, PH, PE, VES, SYS, DMG, AS
27	AP029	JT, AS, TT

Berikut ini adalah frekuensi kemunculan tiap item yang ada pada atribut art pada table 3.5 di bawah ini :

Tabel 3.5 Frekuensi Kemunculan Tiap Art

Art	Frekuensi
AS	6
AD	2
AM	2
BLD	1

Art	Frekuensi
MC	3
MF	5
MFD	3
NIN	1

Art	Frekuensi
CK	2
CE	5
CL	5
CT	3
DMD	6
DMG	5
DA	1
FAI	1
FI	1
FD	4
GI	5
HL	2
JT	3
LR	1
LN	1
MCH	1
MM	1

Art	Frekuensi
NE	2
PE	5
PX	1
PU	3
PH	3
RTS	3
SC	5
SH	1
SYS	2
TT	4
TA	2
TS	3
TON	1
TRI	1
VES	3
WE	2
RTS	**3**

3. Setelah frekuensi didapatkan untuk tiap item maka pemberian prioritas untuk setiap item dilakukan untuk memulai proses pembentukan *tree,* hasil pengurutan berdasarkan frekuensi tersebut akan dijadikan sebagai urutan kemunculan item dalam setiap transaksi yang mengacu pada pemberian prioritas yang ditunjukan pada tabel di bawah ini :

Tabel 3.6 Pengurutan Berdasarkan Frekuensi dan ditentukan nilai support = 3

Art	Frekuensi
AS	6
DMD	6
CE	5
CL	5
DMG	5
SC	5
GI	5
PE	5
MF	5
FD	4
TT	4
CT	3
JT	3
MC	3
PU	3
PH	3
TS	3
MFD	3
VES	3

Tabel 3.7 Pemberian Skala Prioritas

Art	Frekuensi	Prioritas
AS	6	1
DMD	6	2
CE	5	3
CL	5	4
DMG	5	5
SC	5	6

Art	Frekuensi	Prioritas
GI	5	7
PE	5	8
MF	5	9
FD	4	10
TT	4	11
CT	3	12
JT	3	13
MC	3	14
PU	3	15
PH	3	16
TS	3	17
MFD	3	18
VES	3	19
RTS	3	20

4. Setelah data diberi skala prioritas sesuai urutan secara *descending* pada tabel 3.7, maka selanjutnya adalah mengurutkan item transaksi berdasarkan prioritas yang telah ditentukan sebelumnya. Berikut adalah hasil pengurutan berdasarkan skala priotitas tiap item

Tabel 3.8 Data Yang Sudah Di Urutkan Berdasarkan Prioritas

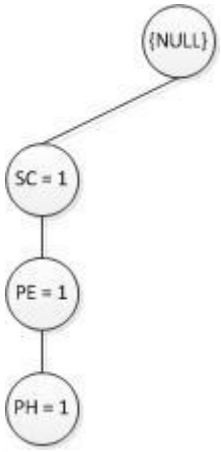
TID	TRANSAKSI	*ORDER ITEM*
1	SC, PH, PE,	SC, PE, PH
2	MC, MF, GI, FD	GI, FD, MF, MC
3	DMG, DMD, CT, CL, CE	DMD, CE, CL, DMG, CT
4	AS	AS
5	MC	MC
6	FD	FD
7	CL	CL

TID	TRANSAKSI	*ORDER ITEM*
8	PU, MFD, CE, RTS	CE, PU, MFD, RTS
9	DMD, MF, GI	DMD, GI, MF
10	PU, TT, TS, MFD	TT, PU, TS, MFD
11	MC, SC, PH, MF	SC, MF, MC, PH
12	GI, FD, DMG, DMD, CE	DMD, CE, DMG, GI, FD
13	CT, AS, VES	AS, CT, VES
14	GI	GI
15	SC, DMD	DMD, SC
16	DMG, MF, AS	AS, DMG, MF
17	FD, VES,PE	PE, FD, VES
18	CL, CE	CE, CL
19	TT, TS, JT	TT, JT, TS
20	RTS	RTS
21	PE, MFD, DMG, CL, AS	AS, CL, DMG, PE, MFD
22	CE, CL, CT	CE, CL, CT
23	DMD, GI, MF, PE, SC	DMD, SC, GI, PE, MF
24	TS, TT, PU, JT	TT, JT, PU, TS
25	RTS, DMD	DMD, RTS
26	SC, PH, PE, VES, DMG, AS	AS, DMG, SC, PE, PH, VES
27	JT, AS, TT	AS, TT, JT

5. Setelah pengurutan pada setiap transaksi sesuai item, selanjutnya masuk pada pembentukan FP-Tree. Berikut adalah tahap-tahap dalam pembuatan FP-Tree yaitu:

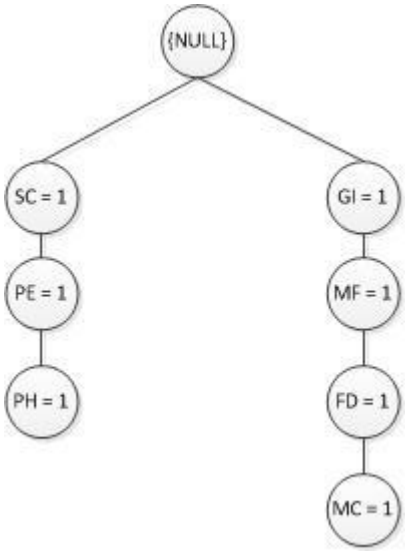
   a. Lakukan pembacaan pada data pertama yang akan diproses.
   b. Lakukan pengecekan pada *root*, jika *root* belum mempunyai anak (*child*), maka data yang dibaca tadi menjadi anak dari *root*. Dan *support count* dari masing-masing *node* bertambah 1, kecuali *root.*

c. 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.
d. Lakukan pengecekan pada *root*, jika *root* sudah mempunyai anak, maka data yang dibaca atau data yang keluar berbeda dengan data yang kelaur sebelumnya maka itu menjadi anak terakhir dari *root* dan menjadi sibling dari *node* data pertama. Dan *support count* dari *node* tersebut bertambah 1.
e. Lakukan langkah tersebut hingga semua data yang keluar.
f. Jika ada data yang sama setelah pembacaan, namun berbeda *lintasan* maka *nod*e pada *laintasan* yang lain tidak bertambah *support* count-nya, hanya data yang dibaca saja yang bertambah *support count*-nya. Pembentukan FP-Tree lengkap dapat dilihat pada lampiran D-4.

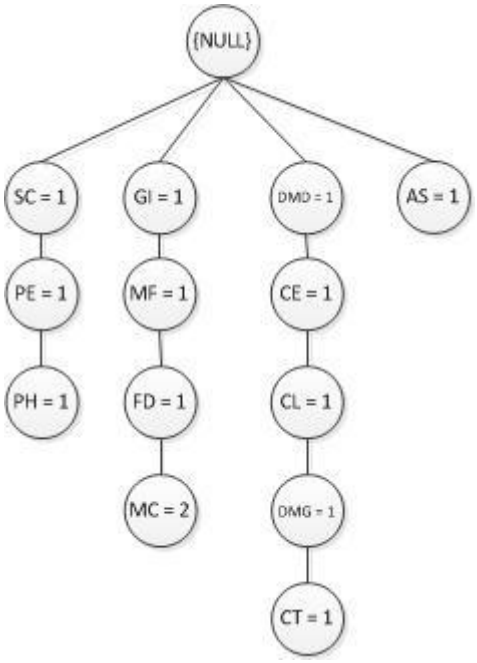


Gambar 3.1 Hasil Pembentukan FP-Tree Setelah TID 1

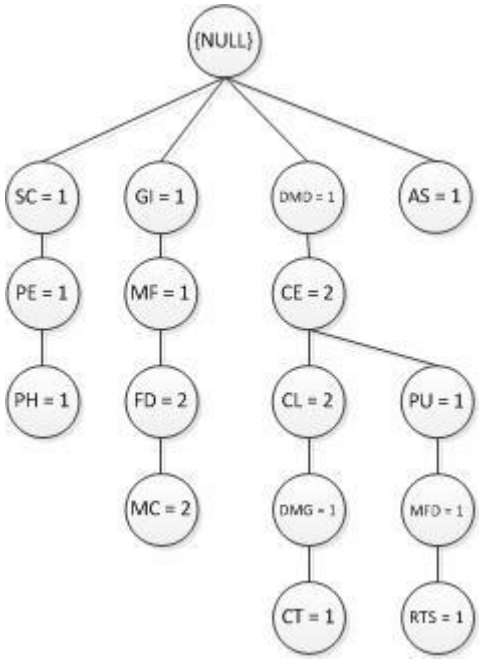
Pembacaan TID 2 yaitu GI, MF, FD, MC. Karena awalan *leaf* GI belum terbentuk maka dibuat jalur baru seperti pada gambar 3.2 di bawah ini.



Gambar 3.2 Hasil Pembentukan FP-Tree Setelah TID 2

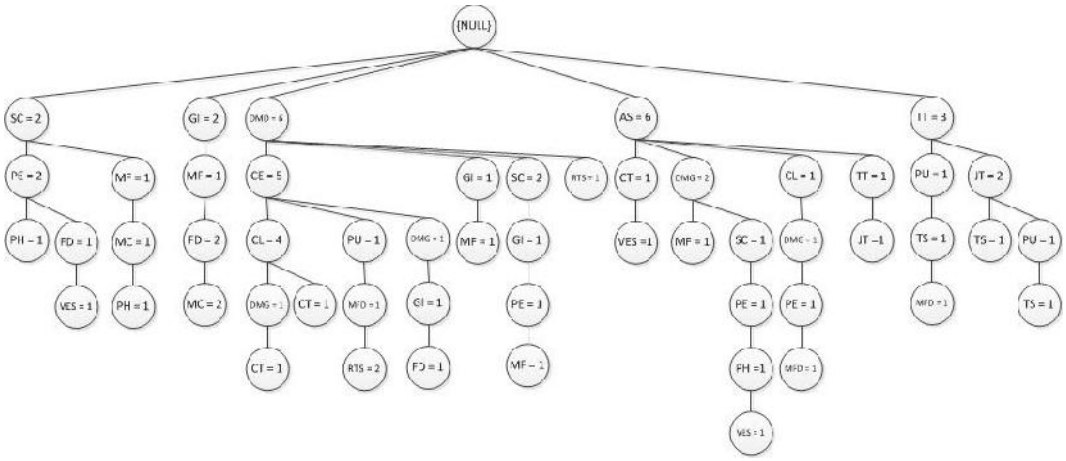


Gambar 3.3 Hasil Pembentukan FP-Tree Setelah TID 5



Gambar 3.4 Hasil Pembentukan FP-Tree Setelah TID 8

Setelah proses pembacaan TID 27 yaitu AS, TT, JT, maka terbentuklah *tree* dari semua transaksi yang akan digunakan seperti pada gambar 3.5 dibawah ini.



Gambar 3.5 Hasil Pembentukan FP-Tree Setelah TID 27

6. Setelah tahap pembangunan FP-Tree dari sekumpulan data transaksi dilakukan, maka akan diterapkan algoritma FP-Growth untuk proses pencarian *frequent itemset* yang signifikan. Berikut langkah-langkah utama algoritma FP-Growth, yaitu :

a. Tahap pembangkitan *conditional pattern base*

Pembangkitan *conditional pattern base* didapatkan melalui hasil FP-Tree seluruhnya, dengan mencari *support count* terkecil sesuai dengan hasil pengurutan *priority.* Dan telah didapatkan yaitu : VES, MFD, PH, CT, MF, RTS, MC, PE, GI, DMG, TS, PU, JT, CL, TT, CE, SC, DMD dan AS. Berikut adalah hasil tabel dari *conditional pattern base :*

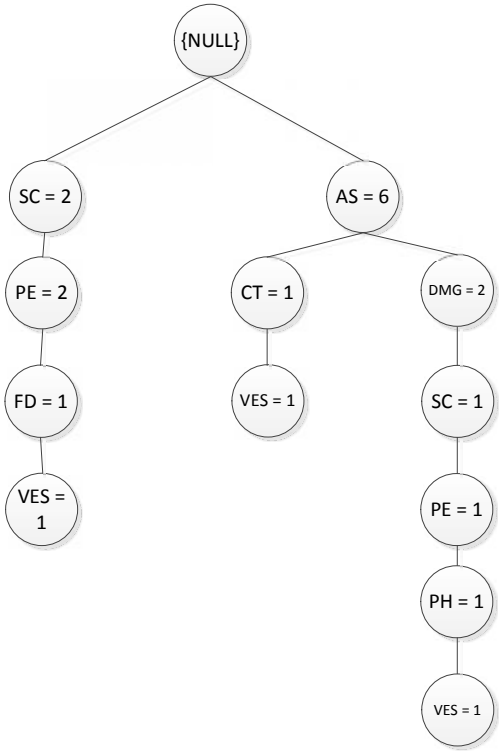
Tabel 3.9 *Conditional Pattern base*

Items	***Conditional Pattren Base***
VES	{ FD:1, PE:1, SC:1 },{ CT:1, AS:1 },{ PH:1, PE:1, SC:1, DMG:1, AS:1}
MFD	{ PU:1, CE:1, DMD:1},{ PE:1, DMG:1, CL:1, AS:1},{ TS:1, PU:1, TT:1}
PH	{ PE:1, SC:1},{ MC:1, SC:1},{ PE:1, SC:1, DMG:1, AS:1}
CT	{ DMG:1, CL:1, CE:1, DMD:1},{ CL:1, CE:1, DMD:1},{AS:1}
MF	{ SC:1},{GI}, { GI:1, DMD:1 },{ PE:1, GI:1, SC:1, DMD:1},{DMG:1, AS:1 }
FD	{ PE:1, SC:1 },{MF,GI:2},{ GI:1, MF:1, CE:1, DMD:1}
RTS	{ MFD:2, PU:2, CE:2, DMD:2}, {DMD:1}
MC	{SC:1},{ MF:2, FD:2, GI: 2}
PE	{SC:2}.{ GI:1, SC:1, DMD:1 },{ SC:1, DMG:1, AS:1 },{ DMG:1, CL:1, AS:1 }
GI	{NULL},{ MF:1, CE:1, DMD:1 },{DMD:1},{ SC:1, DMD:1}
DMG	{ CL:1, CE:1, DMD:1 },{AS:2},{ CL:1, AS:1 }
TS	{ PU:1, TT:1 }, { JT:1, TT:1 },{ PU:1, JT:1, TT:1}
PU	{ CE:1, DMD:1 },{TT:1},{ JT:1, TT:1 }
JT	{ TT:1, AS:1 },{TT:2}
CL	{ CE:4, DMD: 4 },{AS:1}
TT	{AS:1},{NULL}
CE	{ DMD:5}
SC	{NULL},{DMD:2},{ DMG :1, AS:1}

b. Tahap pembangkitan *conditional* FP-Tree

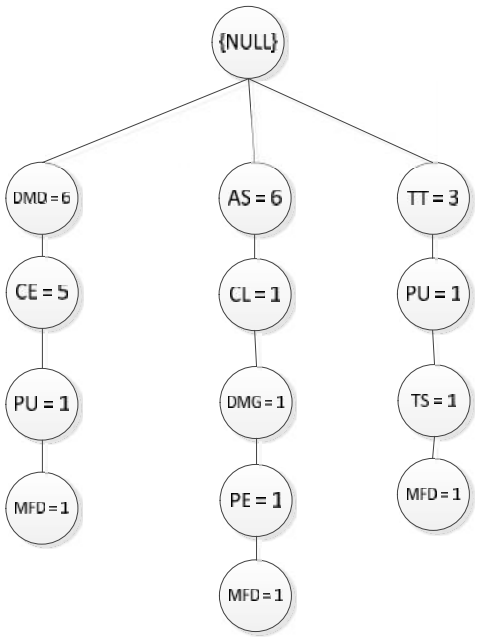
Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan minimum *support count* akan dibangkitkan dengan *conditional* FP-Tree. Tahap pembentukan *Conditional* FP-Tree untuk *item* VES dengan *Conditional Pattern Base* {SC,PE,FD:1}, {AS,CT:1}, {AS,DMG,SC,PE,PH:1} yaitu . null

karena walaupun memenuhi *support count* ≥ 3 dan frequent, tapi pada proses pembangkitan tidak ada yang memenuhi minimum support count.



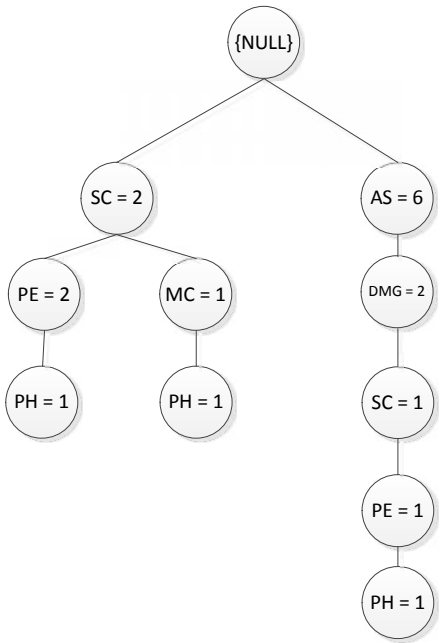
Gambar 3.6 Lintasan Mengandung Simpul VES

Conditional FP-Tree untuk *item* MFD dengan *Conditional pattern base* { PU:1, CE:1, DMD:1},{ PE:1, DMG:1, CL:1, AS:1},{ TS:1, PU:1, TT:1} yaitu *null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambar 3.7 Lintasan Mengandung Simpul MFD

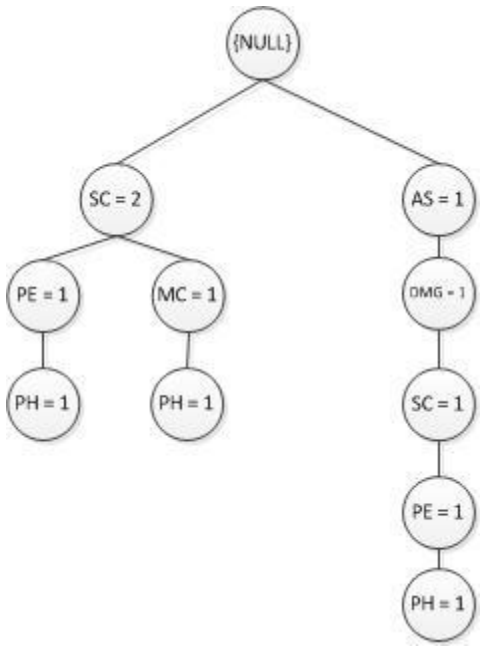
Conditional FP-Tree untuk *item* PH dengan *Conditional Pattern Base* {SC,PE:1},{SC,MC:1},{AS,DMG,SC,PE:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.8 *Lintasan Mengandung Simpul PH*

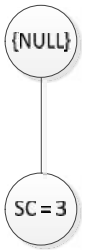
Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk PH, ada 3 lintasan yaitu :
     - Lintasan pertama : {SC,PE,PH:1}
     - Lintasan kedua : {SC,MC, PH:1}
     - Lintasan ketiga : {AS,DMG,SC,PE,PH:1}
   - Pada lintasan yang berakhiran {PH:1} dengan *support count* adalah 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran PH, akan disamakan dengan *support count* PH yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



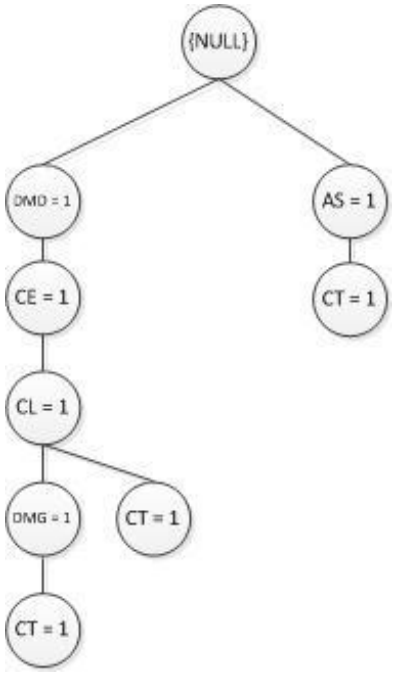
Gambar 3.9 **Conditional FP-Tree *PH Setelah Dibangkitkan***

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. SC:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {PH, SC:3}



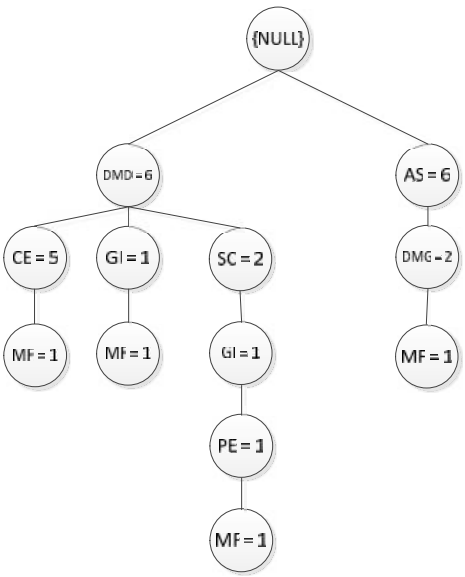
Gambar 3.10 Conditional FP-Tree *PH,* SC:3 *Hasil* Frequent Itemset

Conditional FP-Tree untuk *item* CT dengan *Conditional pattern base* { DMG:1, CL:1, CE:1, DMD:1},{ CL:1, CE:1, DMD:1},{AS:1} yaitu *null* karena walaupun memenuhi *support count* ≥ *3* dan *frequent,* tapi pada proses pembangkitan tidak ada yang memenuhi minimum *support count.*



Gambar 3.11 ***Lintasan Mengandung Simpul CT***

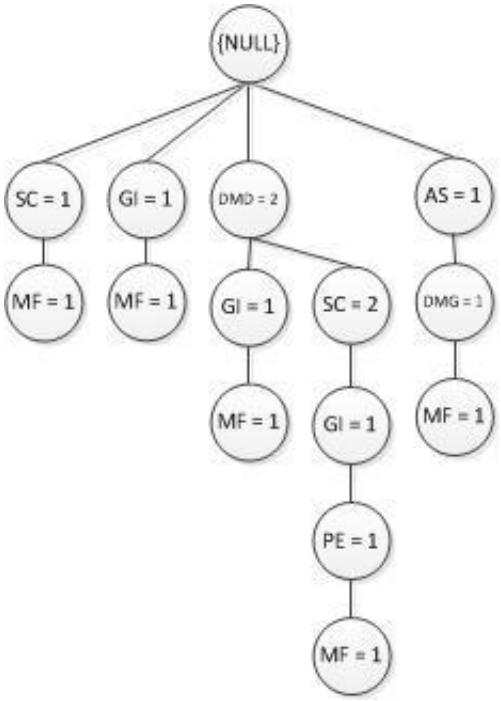
Conditional FP-Tree untuk *item* MF dengan *Conditional Pattern Base* {SC:1}, {GI:1}, {DMD,GI:1}, {DMD,SC,GI,PE:1}, {AS,DMG:1} yang setelah dijumlahkan berdasarkan *support count* =*5* telah memenuhi minimum *support* ≥ *3.*



Gambar 3.12 ***Lintasan Mengandung Simpul MF***

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk MF, ada 5 lintasan yaitu :
     - Lintasan pertama : {SC, MF:1}
     - Lintasan kedua : {GI, MF:1}
     - Lintasan ketiga : {DMD, GI, MF:1}
     - Lintasan keempat : {DMD, SC, GI, PE, MF:1}
     - Lintasan kelima : {AS,DMG, MF:1}
   - Pada lintasan yang berakhiran {MF:1} dengan *support count* adalah 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran MF, akan disamakan dengan *support count* MF yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.
   -



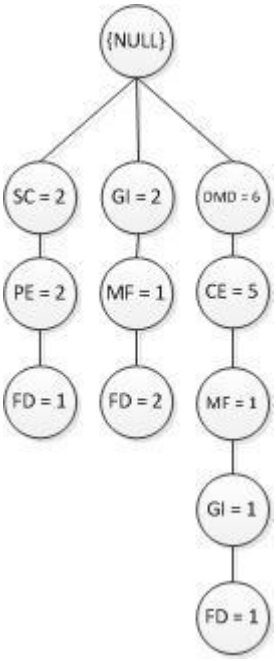
Gambar 3.13 Conditional FP-Tree *MF Setelah Dibangkitkan*

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {PH, GI:3}



Gambar 3.14 Conditional FP-Tree MF, GI:3 Hasil *Frequent Itemset*

Conditional FP-Tree untuk *item* FD dengan *Conditional pattern base* { PE:1, SC:1 },{GI:2},{ GI:1, MF:1, CE:1, DMD:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.15 Lintasan Mengandung Simpul FD

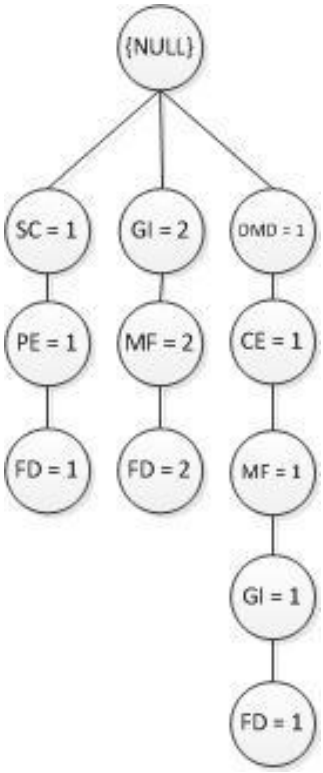
Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.

   -Dalam *conditional* FP-Tree untuk FD, ada 3 lintasan yaitu :

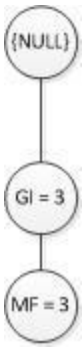
   - Lintasan pertama : {SC, PE, FD:1}
   - Lintasan kedua : { GI, MF, FD:2}
   - Lintasan ketiga : {DMD, CE, MF, GI, FD:1}

- Pada lintasan yang berakhiran {FD:1} dan{FD:2} dengan *support count* adalah 1,2 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran MF, akan disamakan dengan *support count* FD yaitu sesuai dengan *support count* yang dimilikinya.
- Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



Gambar 3.16 *Conditional* FP-Tree FD Setelah Dibangkitkan

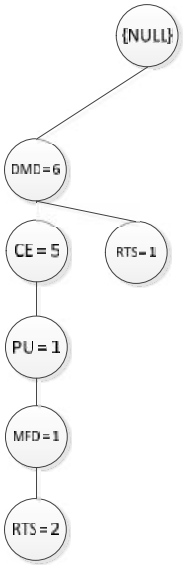
2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {FD, GI, MF:3}



Gambar 3.17 *Conditional* FP-Tree *FD,* GI:3 dan MF:3 Hasil *Frequent Itemset*

dan termasuk ke dalam *item* yang *frequent* yaitu FD : {GI: 3} :

- Tahap pembentukan *Conditional* FP-Tree untuk *item* RTS dengan *Conditional Pattern Base* {DMD,CE,PU,MFD:2}, {DMD:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*

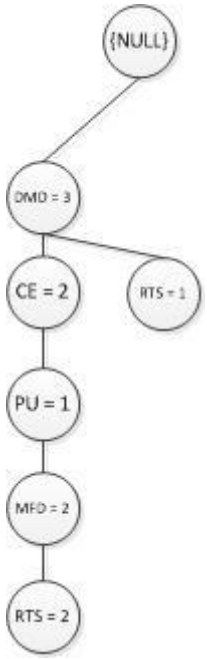


Gambar 3.18 Lintasan Mengandung Simpul RTS

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk RTS, ada 2 lintasan yaitu:
     - Lintasan pertama : {DMD, CE, PU, MFD, RTS:2}
     - Lintasan kedua : {DMD, RTS:1}
   - Pada lintasan yang berakhiran {RTS:2} dan {RTS:1} dengan *support count* adalah 2dan 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran RTS, akan disamakan dengan *support count* RTS yaitu sesuai dengan *support count* yang dimilikinya.

- Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



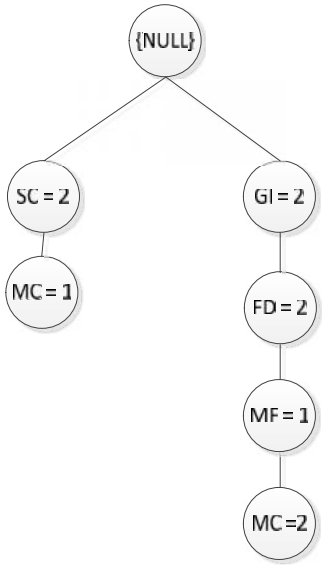
Gambar 3.19 *Conditional* FP-Tree RTS Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {RTS, DMD:3}



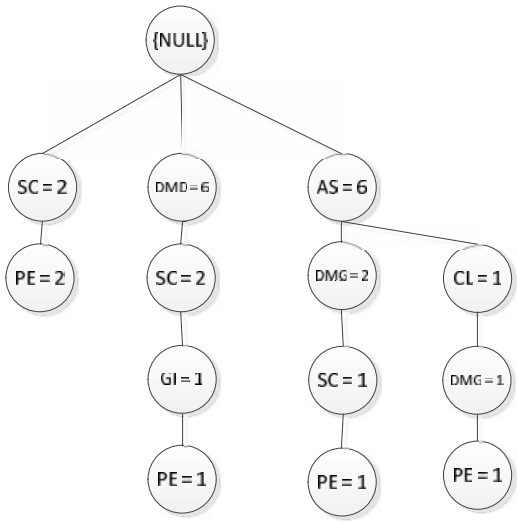
Gambar 3.20 *Conditional* FP-Tree RTS, DMD:3 Hasil *Frequent Itemset*

- *Conditional* FP-Tree untuk *item* MC dengan *Conditional pattern base* {SC:1},{ MF:2, FD:2, GI: 2} yaitu *null* karena tidak memenuhi *support count* ≥ *3* dan *frequent,* tapi pada proses pembangkitan tidak ada yang memenuhi minimum *support count.*



Gambar 3.21 Lintasan Mengandung Simpul MC

- *Conditional* FP-Tree untuk *item* PE dengan *Conditional Pattern Base* {SC:2}.{DMD,SC,GI:1}, {AS,DMG,SC:1}, {AS,CL,DMG:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.22 Lintasan Mengandung Simpul PE

Berikut adalah tahap-tahapannya :

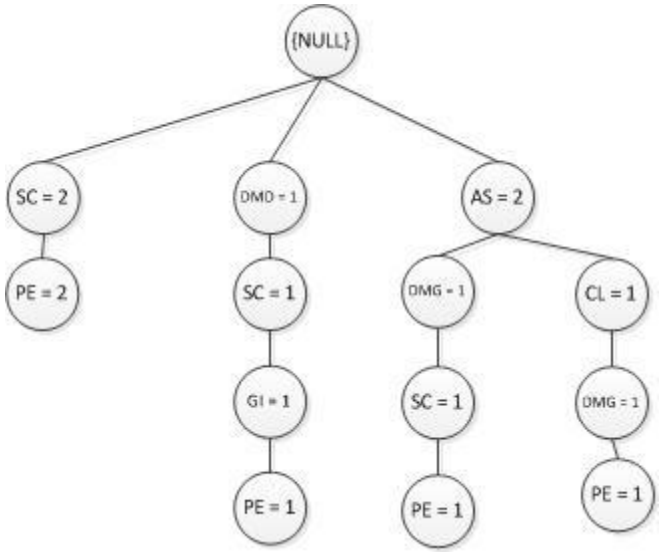
1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.

   -Dalam *conditional* FP-Tree untuk PE, ada 4 lintasan yaitu :

   - Lintasan pertama : {SC, PE:2}
   - Lintasan kedua : {DMD, SC,GI,PE:1}
   - Lintasan ketiga : {AS, DMG,SC,PE:1}
   - Lintasan keempat : {AS,CL,DMG,PE:1}

   -Pada lintasan yang berakhiran {PE:2},{PE:1} dengan *support count* adalah 2 dan 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran PE, akan disamakan dengan *support count* PE yaitu sesuai dengan *support count* yang dimilikinya.

   -Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



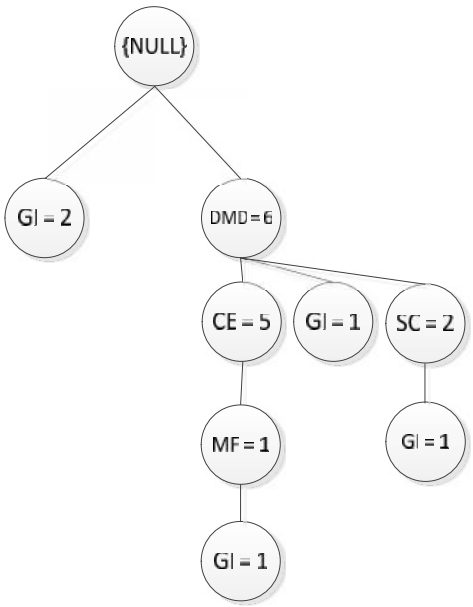
Gambar 3.23 *Conditional* FP-Tree PE Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. SC:4 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {PE, SC:4}



Gambar 3.24 *Conditional* FP-Tree PE, SC:3 Hasil *Frequent Itemset*

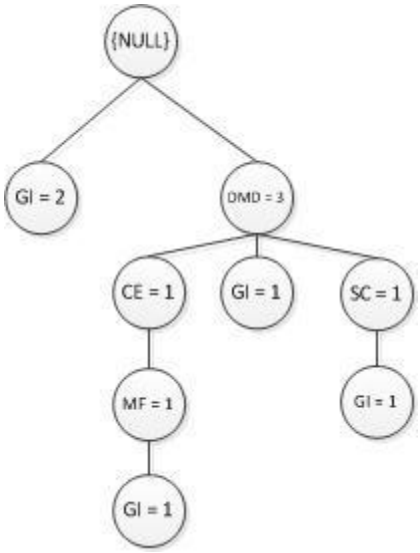
- *Conditional* FP-Tree untuk *item* GI dengan *Conditional Pattern Base* {NULL},{DMD,CE,MF:1},{DMD:1},{DMD,SC:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.25 Lintasan Mengandung Simpul GI

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk PH, ada 4 lintasan yaitu :
     - Lintasan pertama : { GI:2}
     - Lintasan kedua : {DMD, CE, MF, GI:1}
     - Lintasan ketiga : {DMD,GI:1}
     - Lintasan keempat : {DMD,SC, GI:1}
   - Pada lintasan yang berakhiran {GI:1} dengan *support count* adalah 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran GI, akan disamakan dengan *support count* GI yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



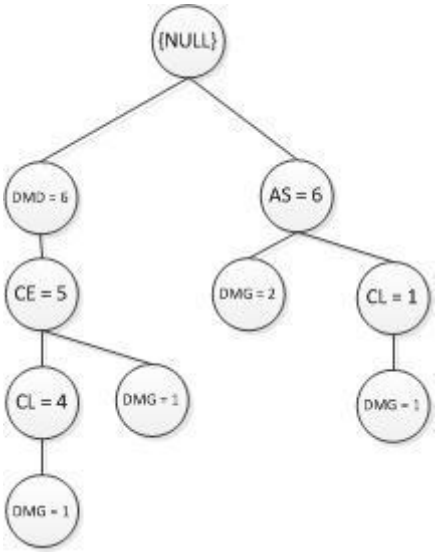
Gambar 3.26 *Conditional* FP-Tree GI Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {GI, DMD:3}



Gambar 3.27 *Conditional* FP-Tree GI, DMD:3 Hasil *Frequent Itemset*

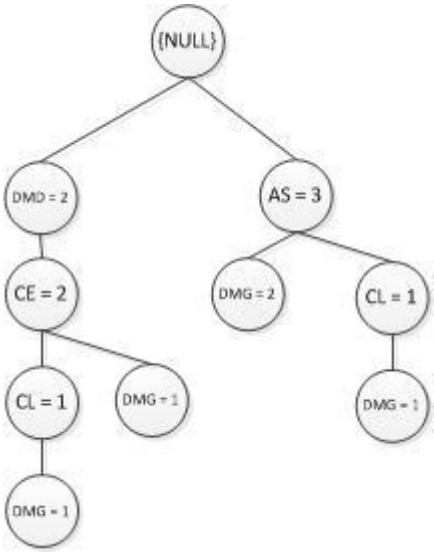
Conditional FP-Tree untuk *item* DMG dengan *Conditional Pattern Base* {DMD,CE,CL},{DMD,CE},{AS:2},{AS,CL:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.28 Lintasan Mengandung Simpul DMG

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk DMG, ada 4 lintasan yaitu:
     - Lintasan pertama : {DMD, CE, CL, DMG:1}
     - Lintasan kedua : {DMD, CE, DMG:1}
     - Lintasan ketiga : {AS, DMG:2}
     - Lintasan empat : {AS, CL, DMG:1}
   - Pada lintasan yang berakhiran {DMG:1} dan {DMG:2} dengan *support count* adalah 1 dan 2 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran DMG, akan disamakan dengan *support count* DMG yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



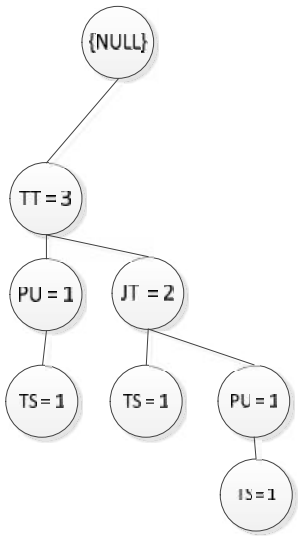
Gambar 3.29 *Conditional* FP-Tree DMG Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. AS:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {DMG, AS:3}



Gambar 3.30 *Conditional* FP-Tree DMG, AS:3 Hasil *Frequent Itemset*

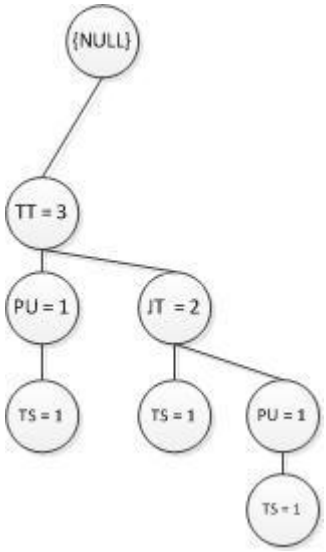
Conditional FP-Tree untuk *item* TS dengan *Conditional Pattern Base* { PU:1, TT:1 }, { JT:1, TT:1 },{ PU:1, JT:1, TT:1} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3*



Gambar 3.31 Lintasan Mengandung Simpul TS

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk TS, ada 3 lintasan yaitu :
     - Lintasan pertama : {TT,PU,TS:1}
     - Lintasan kedua : {TT,JT,TS:1}
     - Lintasan ketiga : {TT,JT,PU,TS:1}
   - Pada lintasan yang berakhiran {TS:1} dengan *support count* adalah 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran TS, akan disamakan dengan *support count* TS yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



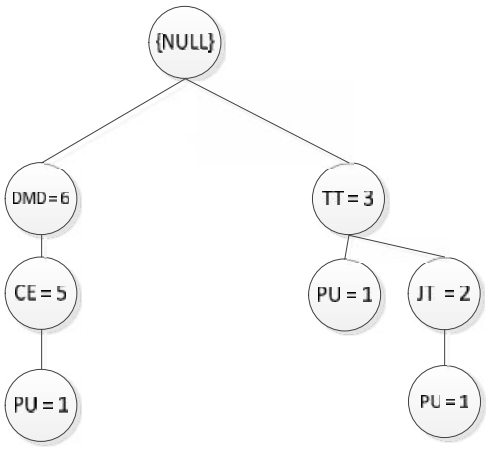
Gambar 3.32 *Conditional* FP-Tree TS Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. TT:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {TS, TT:3}



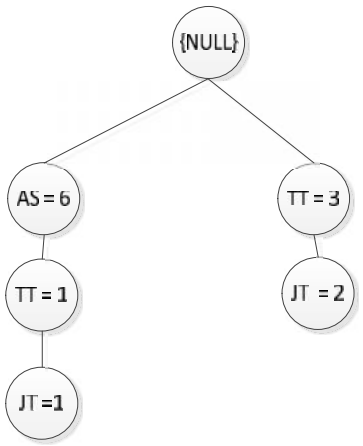
Gambar 3.33 *Conditional* FP-Tree TS, TT:3 Hasil *Frequent Itemset*

- *Conditional* FP-Tree untuk *item* PU dengan *Conditional pattern base* { CE:1, DMD:1 },{TT:1},{ JT:1, TT:1 } yaitu *null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambar 3.34 Lintasan Mengandung Simpul PU

- *Conditional* FP-Tree untuk *item* JT dengan *Conditional Pattern Base* {AS,TT:1},{TT:2} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



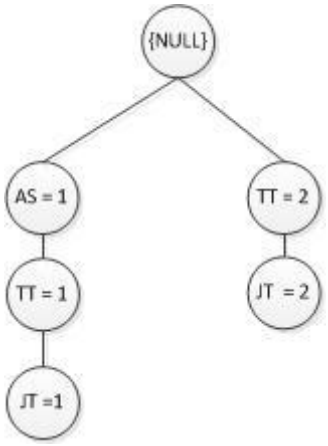
Gambar 3.35 Lintasan Mengandung Simpul JT

Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk JT, ada 2 lintasan yaitu :
     - Lintasan pertama : {AS,TT,JT:1}
     - Lintasan kedua : {TT,JT:2}
   - Pada lintasan yang berakhiran {JT:1},{JT:2} dengan *support count* adalah 1 dan 2 akan dihilangkan dan *support count* yang

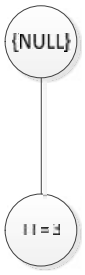
melewati lintasan pertama yang berakhiran JT, akan disamakan dengan *support count* JT yaitu sesuai dengan *support count* yang dimilikinya.

- Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



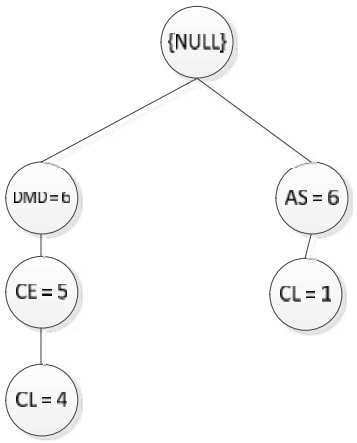
Gambar 3.36 *Conditional* FP-Tree JT Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. TT:3 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {JT, TT:3}



Gambar 3.37 *Conditional* FP-Tree JT, TT:3 Hasil *Frequent Itemset*

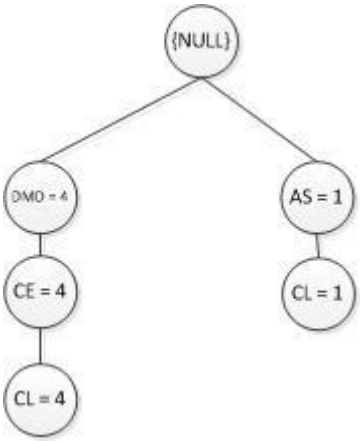
- Tahap pembentukan *Conditional* FP-Tree untuk *item* CL dengan *Conditional Pattern Base* {DMD,CE:4},{AS:1} yang setelah dijumlahkan berdasarkan *support count* = 5 telah memenuhi minimum *support* ≥ *3* .



Gambar 3.38 Lintasan Mengandung Simpul CL

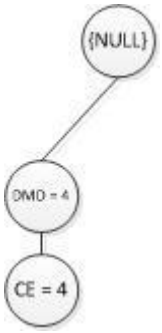
Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk PH, ada 2 lintasan yaitu :
     - Lintasan pertama : {DMD,CE,CL:4}
     - Lintasan kedua : {AS,CL:1}
   - Pada lintasan yang berakhiran {CL:4} dan {CL:1} dengan *support count* adalah 4 dan 1 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran CL, akan disamakan dengan *support count* CL yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



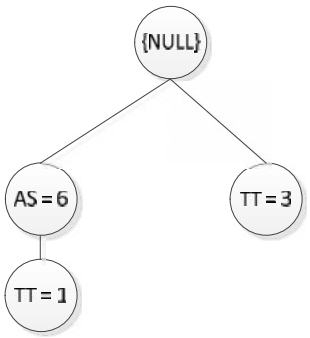
Gambar 3.39 *Conditional* FP-Tree CL Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD, CE:4 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {CL, DMD,CE:4}



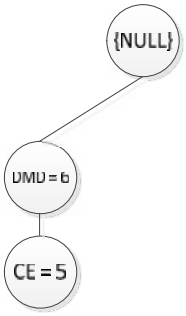
Gambar 3.40 *Conditional* FP-Tree CL, DMD,CE:4 Hasil *Frequent Itemset*

- *Conditional* FP-Tree untuk *item* TT dengan *Conditional pattern base* {AS:1},{NULL} yaitu *null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambar 3.41 Lintasan Mengandung Simpul TT

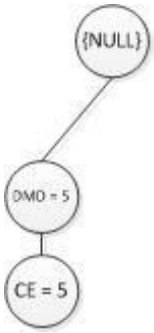
- Tahap pembentukan *Conditional* FP-Tree untuk *item* CE dengan *Conditional Pattern Base* { DMD:5} yang setelah dijumlahkan berdasarkan *support count* = 3 telah memenuhi minimum *support* ≥ *3.*



Gambar 3.42 Lintasan Mengandung Simpul CE

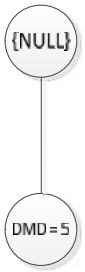
Berikut adalah tahap-tahapannya :

1. Jumlah *support count* yang memenuhi *minimum support* akan dibangkitkan dengan *conditional* FP-Tree.
   - Dalam *conditional* FP-Tree untuk CE, ada 1 lintasan yaitu :
     - Lintasan keempat : {DMD,CE: 5}
   - Pada lintasan yang berakhiran {CE:5} dengan *support count* adalah 5 akan dihilangkan dan *support count* yang melewati lintasan pertama yang berakhiran CE, akan disamakan dengan *support count* CE yaitu sesuai dengan *support count* yang dimilikinya.
   - Proses selanjutnya adalah apabila dalam suatu lintasan, melintasi dua atau lebih, maka *support count*-nya akan selalu bertambah 1.



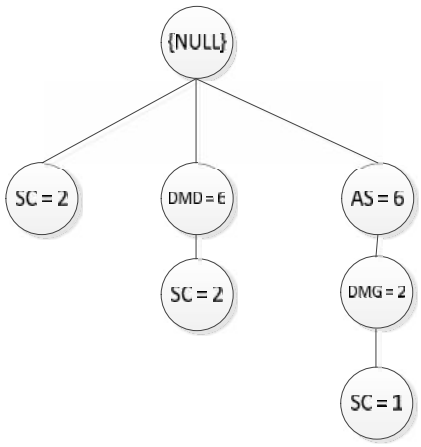
Gambar 3.43 *Conditional* FP-Tree CE Setelah Dibangkitkan

2. Setelah proses pembangkitan, maka proses terakhir adalah mencari *frequent itemset* yang keluar yang memenuhi *minimum support*. DMD:5 adalah satu-satunya yang keluar yang dapat memenuhi *minimum.* Dan *frequent itemset* yang didapatkan adalah {CE, DMD:5}



Gambar 3.44 *Conditional* FP-Tree CE, DMD:5 Hasil *Frequent Itemset*

Conditional FP-Tree untuk *item* SC dengan *Conditional pattern base* {NULL},{DMD:2},{ DMG :1, AS:1} yaitu *null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambar 3.45 Lintasan Mengandung Simpul SC

- *Conditional* FP-Tree untuk *item* DMD dengan *Conditional pattern base null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambar 3.46 Lintasan Mengandung Simpul DMD

- *Conditional* FP-Tree untuk *item* AS dengan *Conditional pattern base null* karena tidak memenuhi *support count* ≥ *3* dan tidak termasuk ke dalam *item* yang *frequent.*



Gambra 3.47 Lintasan Mengandung Simpul AS

c. Tahap pencarian *frequent itemset.*

Setelah mendapatkan hasil dari *conditional* FP-Tree dari table *conditional pattern base,*ditemukan beberapa *frequent itemset* yang keluar. Hasil dapat dilihat pada tabel 3.10 di bawah ini:

Tabel 3.10 Hasil *Conditional* FP-Tree

Items	***Conditional* FP-Tree**
VES	NULL
MFD	NULL
PH	{SC:3}
CT	NULL
MF	{GI:3}
FD	{GI,MF:3}
RTS	{DMD:3}
MC	NULL
PE	{SC:4}
GI	{DMD:3}
DMG	{AS:3}
TS	{TT:3}
PU	NULL
JT	{TT:3}
CL	{DMD,CE:4}
TT	NULL
CE	{DMD:5}
SC	NULL
DMD	NULL
AS	NULL

Setelah membuat *conditional pattern base,* selanjutnya membuat *conditional* FP-Tree, untuk setiap *pattren base* dilakukan hal berikut :

a. Akumulasikan *count* untuk setiap *item* pada dalam tabel 3.8

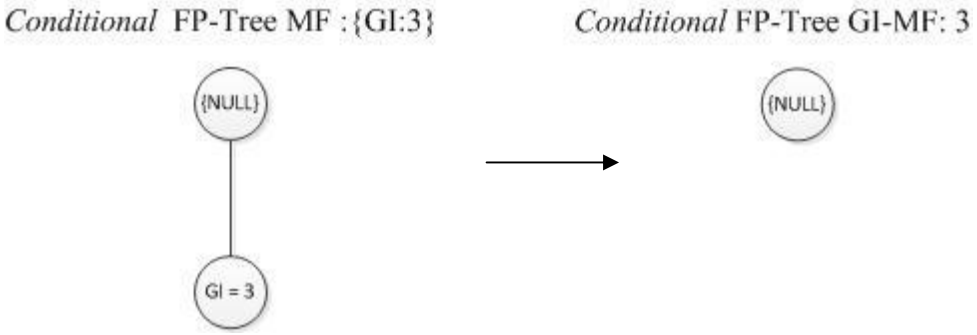
b. Buat *conditional* FP-Tree untuk pembangkitan *frequent item* dengan menggunakan data pada tabel 3.8 di atas.

- Untuk item PH, *item frequent itemset* yang keluar adalah {PH,SC: 3}



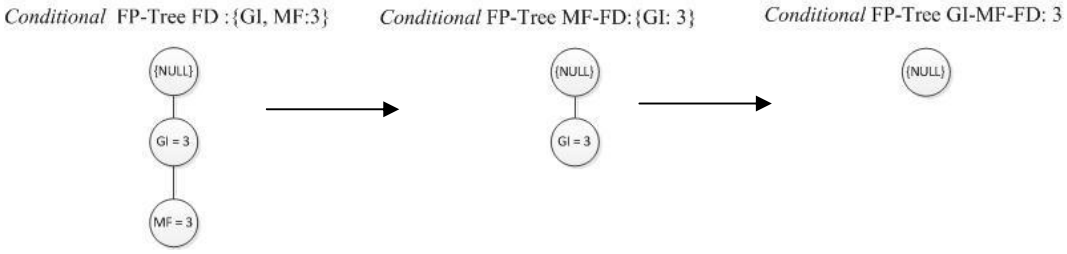
Gambar 3.48 *Conditional* FP-Tree untuk *item* PH

- Untuk item MF, *item frequent itemset* yanng keluar adalah {MF,DMD:3}



Gambar 3.49 *Conditional* FP-Tree untuk *item* MF

- Untuk item FD, *item frequent itemset* yang keluar adalah {FD,GI:3}



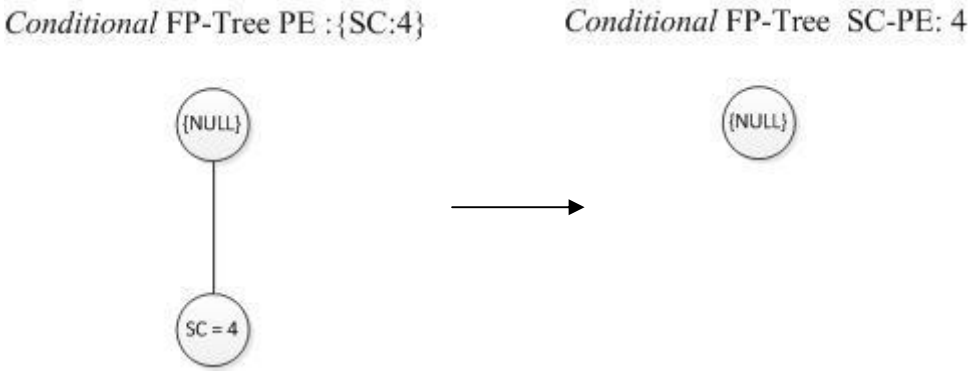
Gambar 3.50 *Conditional* FP-Tree untuk *item* FD

- Untuk item RTS, *item frequent itemset* yang keluar adalah {RTS,DMD:3}



Gambar 3.51 *Conditional* FP-Tree untuk *item* RTS

- Untuk item PE, *item frequent itemset* yang keluar adalah {PE,SC:4}



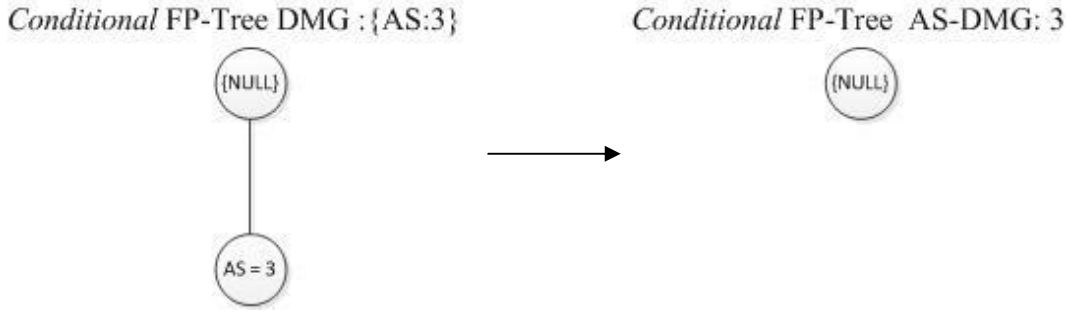
Gambar 3.52 *Conditional* FP-Tree untuk *item* PE

- Untuk item GI, *item frequent itemset* yang keluar adalah {GI,DMD:3}



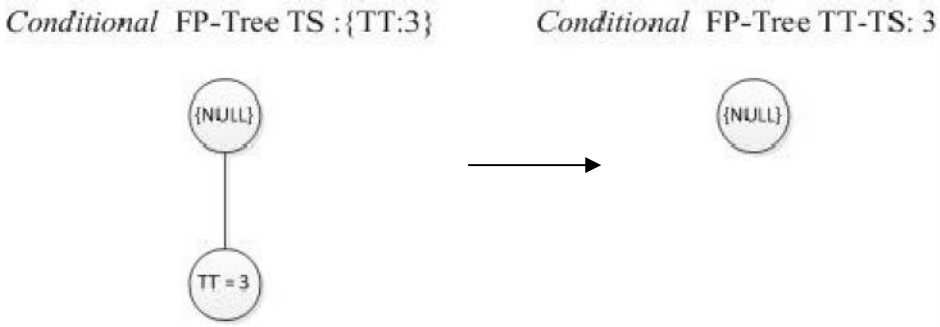
Gambar 3.53 *Conditional* FP-Tree untuk *item* GI

- Untuk item DMG, *item frequent itemset* yang keluar adalah {DMG,AS:3}



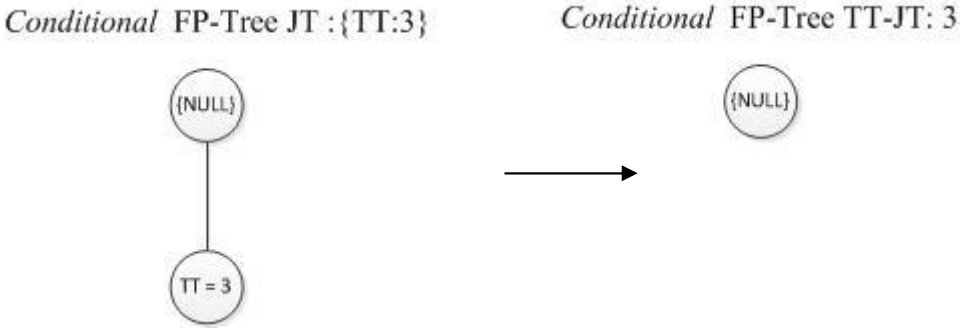
Gambar 3.54 *Conditional* FP-Tree untuk *item* DMG

- Untuk item TS, *item frequent itemset* yang keluar adalah {TS,TT:3}



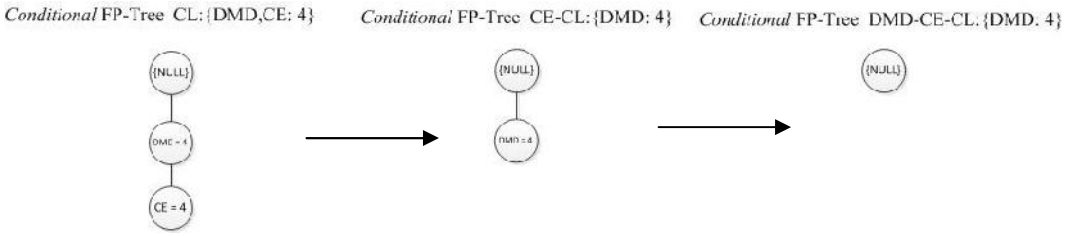
Gambar 3.55 *Conditional* FP-Tree untuk *item* TS

- Untuk item JT, *item frequent itemset* yang keluar adalah {JT,TT:3}



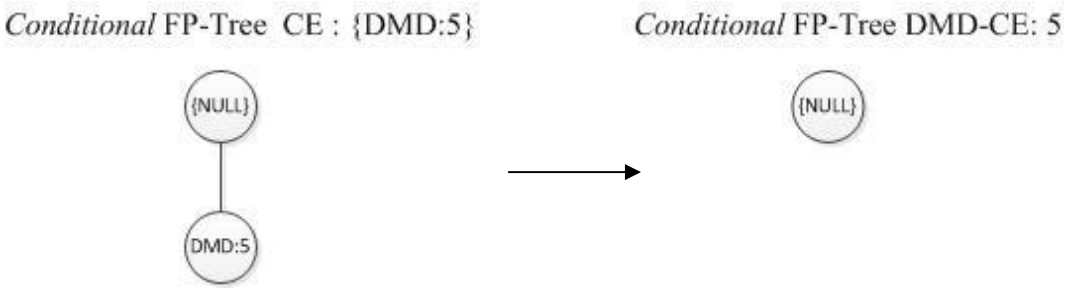
Gambar 3.56 *Conditional* FP-Tree untuk *item* JT

- Untuk item CL, *item frequent itemset* yang keluar adalah {CL,DMD,CE:4}



Gambar 3.57 *Conditional* FP-Tree untuk *item* CL

- Untuk item CE, *item frequent itemset* yang keluar adalah {CE,DMD:5}



Gambar 3.58 *Conditonal* FP-Tree untuk *item* CE

7. Setelah mendapatkan hasil *frequent itemset* dari tiap *conditional* FP-Tree, selanjutnya adalah menentukan *frequent itemset* setiap *suffix*

Tabel 3.11 Hasil *Frequent Itemset*

Suffix	***Frequent Itemset***
PH	PH, SC-PH
MF	MF, GI-MF
FD	FD, MF-FD, GI-MF-FD
RTS	RTS, DMD-RTS
PE	PE, SC-PE
GI	GI, DMD-GI
DMG	DMG, AS-DMG
TS	TS, TT-TS
JT	JT, TT-JT

Suffix	*Frequent Itemset*
CL	CL, CE-CL, DMD-CE-CL
CE	CE, DMD-CE

Dari hasil *frequent Itemset* diatas akan di-*generate* untuk mendapatkan *rule* dengan *minimum support*=3 dan *minimum confidence*=60% dengan masing-masing *itemset* dikombinasikan dengan *itemset* lain. Dengan menggunakan persamaan(2.2) untuk memperoleh nilai *support* untuk 2 *item* serta persamaan(2.4) untuk memperoleh nilai *confidence*,maka *rule* yang sudah didapat akan diberikan nilai *support* dan *confidence* menggunakan persamaan yang sudah disebutkan di atas. Berikut ini adalah hasil dari pemberian nilai untuk *support* dan *confidence* dari tiap *rule.*

berikut merupakan penghitungan untuk setiap *frequent itemset* yang didapatkan dengan menggunakan persamaan (2.2) untuk menghitung *support* dua item dan menghitung nilai *confidence* menggunakan persamaan (2.3) :

$$Support\,(A,B) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi}} \text{x100\%..Persamaan (2.2)}$$

$$Confidence = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi yang mengandung A}} \text{x100\%…..Persamaan(2.3)}$$

Tabel 3.12 Hasil *Generate Rule*

Rule	*Support*	*Confidence* (%)
SC → PH	(3/27)*100 = 11	(3/5)*100 = 60
PH → SC	(3/27)*100 = 11	(3/3)*100 = 100
GI →MF	(3/27)*100 = 11	(3/5)*100 = 60
MF →GI	(3/27)*100 = 11	(3/5)*100 = 60
MF→ FD	(1/27)*100 = 4	(1/5)*100 = 20
FD → MF	(1/27)*100 = 4	(1/4)*100 = 25
GI-MF→ FD	(1/27)*100 = 4	(1/3)*100 = 33.33
FD → MF-FD	(1/27)*100 = 4	(1/4)*100 = 25

Rule	*Support*	*Confidence* (%)
DMD → RTS	(1/27)*100 = 4	(1/6)*100 = 16.67
RTS → DMD	(1/27)*100 = 4	(1/3)*100 = 33.33
SC → PE	(3/27)*100 = 11	(3/5)*100 = 60
PE → SC	(3/27)*100 = 11	(3/5)*100 = 60
DMD →GI	(3/27)*100 = 11	(3/6)*100 = 50
GI → DMD	(3/27)*100 = 11	(3/5)*100 = 60
AS →DMG	(3/27)*100 = 11	(3/6)*100 = 50
DMG → AS	(3/27)*100 = 11	(3/5)*100 = 60
TT →TS	(3/27)*100 = 11	(3/4)*100 = 75
TS → TT	(3/27)*100 = 11	(3/3)*100 = 100
TT →JT	(3/27)*100 = 11	(3/4)*100 = 75
JT → TT	(3/27)*100 = 11	(3/3)*100 = 100
CE →CL	(3/27)*100 = 11	(3/5)*100 = 60
CL → CE	(3/27)*100 = 11	(3/5)*100 = 60
DMD-CE-→ CL	(1/27)*100 = 4	(1/6)*100 = 16.67
CE-CL→ DMD	(1/27)*100 = 4	(1/5)*100 = 20
CL-DMD→ CE	(1/27)*100 = 4	(1/5)*100 = 20
CE-DMD→ CL	(1/27)*100 = 4	(1/5)*100 = 20
CL-CE→ DMD	(1/27)*100 = 4	(1/5)*100 = 20
DMD-CE→ CL	(1/27)*100 = 4	(1/6)*100 = 16.67

Setelah perhitungan *confidence* dari setiap produk diketahui maka selanjutnya dilakukan pengecekan setiap produk jika tidak memenuhi *minimum confidence* produk tersebut akan dihapus untuk menemukan *rule*. Setiap art akan dikembalikan sesuai dengan nama dari masing-masing art agar pembacaan *rule* lebih jelas. Adapun rule yang didapat adalah berikut ini :

Tabel 3.13 Hasil *Rule*

Rule	*Support*	*Confidence* (%)
Scrips → Rplnirish	(3/27)*100 = 11	(3/5)*100 = 60
Rplnirish → Scrips	(3/27)*100 = 11	(3/3)*100 = 100
Girl → Manfield	(3/27)*100 = 11	(3/5)*100 = 60
Manfield →Girl	(3/27)*100 = 11	(3/5)*100 = 60
Scrips → Perspective	(3/27)*100 = 11	(3/5)*100 = 60
Perspective → Scrips	(3/27)*100 = 11	(3/5)*100 = 60

Rule	*Support*	*Confidence* (%)
Girl → Diamond	(3/27)*100 = 11	(3/5)*100 = 60
Diminishing → As 16	(3/27)*100 = 11	(3/5)*100 = 60
Thundertype →Tloseries	(3/27)*100 = 11	(3/4)*100 = 75
Tloseries → Thundertype	(3/27)*100 = 11	(3/3)*100 = 100
Thundertype →Jerseytype	(3/27)*100 = 11	(3/4)*100 = 75
Jerseytype → Thundertype	(3/27)*100 = 11	(3/3)*100 = 100
Cityelit →Clover	(3/27)*100 = 11	(3/5)*100 = 60
Clover → Cityelit	(3/27)*100 = 11	(3/5)*100 = 60

3.1.3 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak adalah kebutuhan-kebutuhan apa saja yang dperlukan untuk membangun sebuah perangkat lunak berdasarkan kebutuhan pengguna. Ada dua bagian dalam spesifikasi kebutuhan perangkat lunak yaitu kebutuhan *non fungsional* dan kebutuhan *fungsional.*

3.2.1 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan *non fungsional* dilakukan untuk mengetahui spesifikasi kebutuhan sistem. Spesifikasi kebutuhan melibatkan analis perangkat keras (*hardware*), analisis perangkat lunak(*software*), analisis pengguna (*user*).

3.2.1.1 Analisis Kebutuhan Perangkat Keras

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

1. Analisis spesifikasi kebutuhan perangkat keras pada sistem yang sedang berjalan di AIRPLANE SYSTM adalah :
   a. *Processor :* Intel Core Duo
   b. *Harddisk :* 160 GB
   c. *Memory :* 2 GB
2. Analisis spesifikasi *minimum* kebutuhan perangkat keras pada sistem yang akan dibangun membutuhkan :
   a. *Processor* berkecepatan minimal 2 Ghz
   b. *Harddisk* 1 GB untuk menyimpan data
   c. *Memory* 512 MB

3. Evaluasi kebutuhan perangkat keras

   Dari data spesifikasi perangkat keras yang dimiliki oleh AIRPLANE SYSTM seperti diatas, perangkat keras yang telah dimiliki sudah cukup untuk dapat menjalankan aplikasi pengelolahan data penjualan yang akan dibuat.

3.2.1.2 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pendukung merupakan hal penting bagi pembangunan perangkat lunak yang sedang dirancang. Perangkat lunak pendukung dalam pembangunan perangkat lunak ini adalah sebagai berikut :

1. Analisis kebutuhan perangkat lunak pada sistem yang sedang berjalan di AIRPLANE SYSTM adalah :
   a. Sistem operasi Windows 7
   b. *Microsoft Office*
2. Analisis spesifikasi kebutuhan perangkat lunak untuk menjalankan sistem yang akan dibangun adalah :
   a. Sistem operasi Windows 7
   b. MySQL sebagai *database server*
   c. Visual studio 2012
3. Evaluasi kebutuhan perangkat lunak

   Berdasarkan analisis spesifikasi perangkat lunak yang ada di AIRPLANE SYSTM, spesifikasi perangkat lunak yang akan digunakan kurang memenuhi spesifikasi perangkat lunak yang dibutuhkan. Kekurangan tersebut dapat diatasi dengan meng- *install* MySQL dan Visual Studio pada komputer yang digunakan.

3.2.1.3 Analisis pengguna

Analisis pengguna dibuat untuk memenuhi siapa saja dan seperti apa saja karakteristik yang dimiliki oleh orang yang dapat menggunakan aplikasi yang akan dibuat. Untuk lebih jelas mengenai karakteristik pengguna, dapat dilihat pada tabel 3.13 berikut ini :

Tabel 3.14 Karakteristik Pengguna

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan	Pengalaman
User	Mengoperasikan aplikasi sesuai dengan tugas yang diberikan	Melakukan kegiatan yang berhubungan dengan isi dari tampilan yang ada pada aplikasi	Minimal Sekolah Menengah Atas (SMA)	Memahami konten yang tersedia pada aplikasi	Minimal mampu mengoprasikan komputer

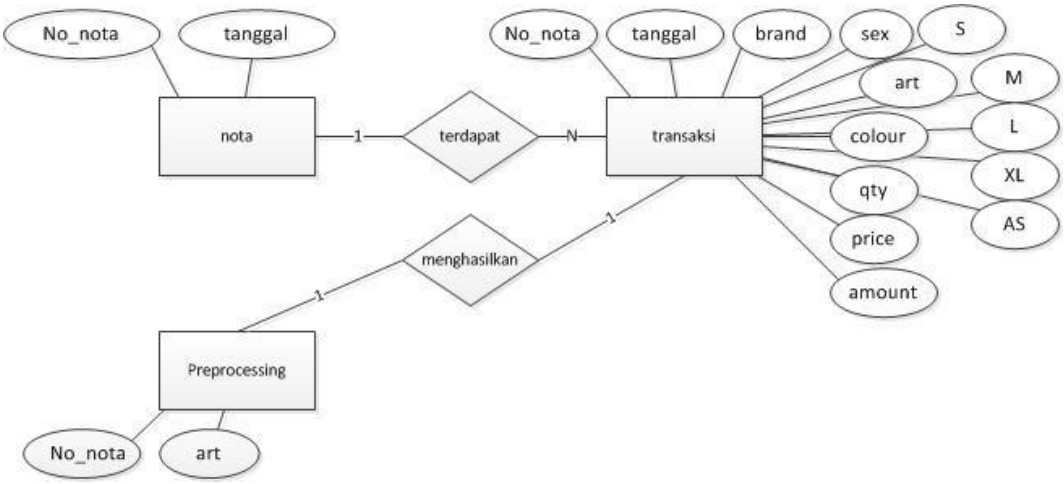
Dengan kondisi yang terdapat diatas, dapat diambil kesimpulan bahwa apliaksi yang dibuat dapat dijalankan oleh staff di perusahaan tersebut. Pegawai Staff dalam hal ini adalah orang yang bekerja di bagian produksi yang akan menggunakan aplikasi yang dibangun.

3.2.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional yang akan dibuat meliputi diagram konteks, Data Flow Diagram (DFD), spesifikasi proses dan kamus data.

3.2.2.1 Analisis Data

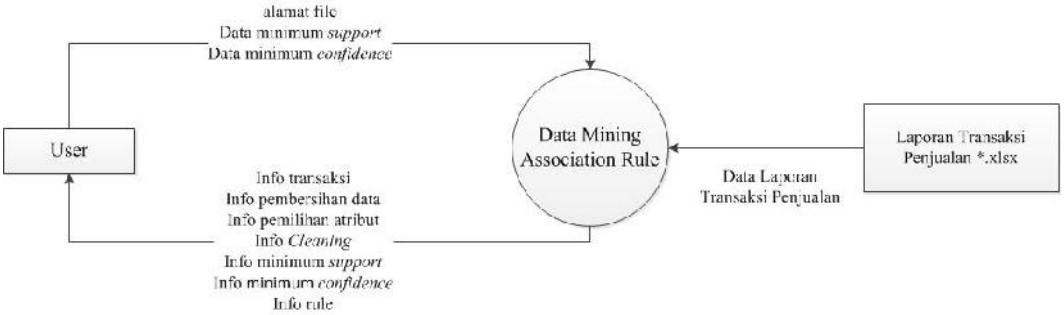
Berikut ini adalah ERD dari perangkat lunak yang akan dibangun.



Gambar 3.59 *Entity Relationship Diagram* (ERD)

3.2.2.2 Diagram konteks

Diagram Konteks adalah diagram yang menggambarkan *input, process* dan *output* pada sistem perangkat lunak yang akan dibangun. Adapun diagram konteks dari sistem yang akan dibangun dapat dilihat pada gambar berikut :



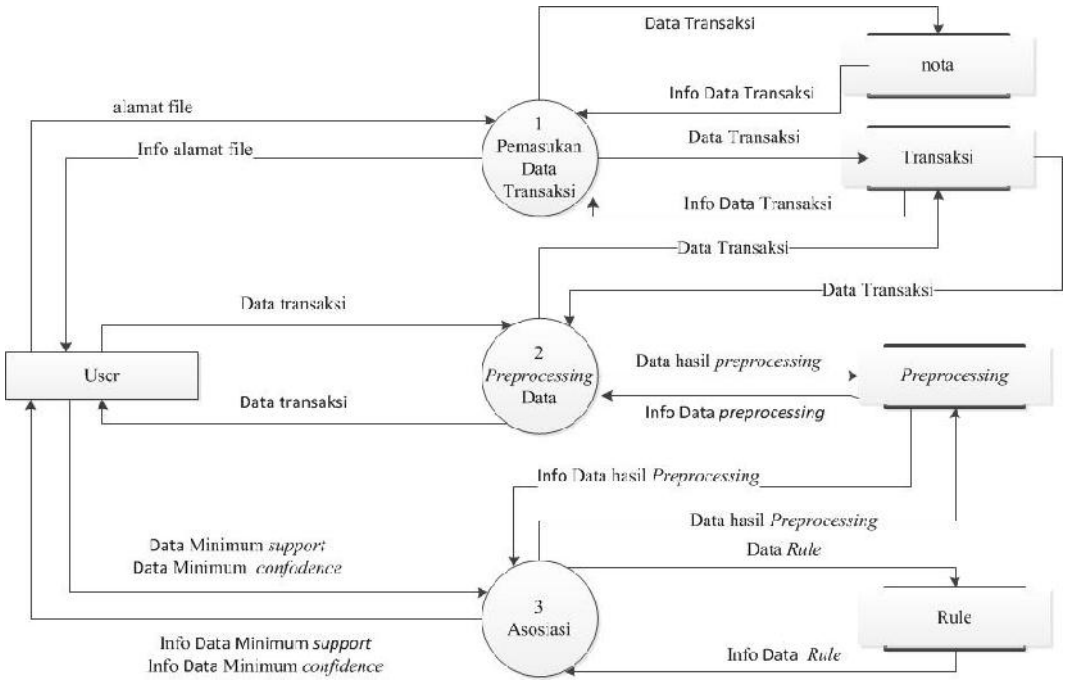
Gambar 3.60 Diagram Konteks

3.2.2.3 Data Flow Diagram

Data Flow Diagram (DFD) merupakan model logika atau proses yang menggambarkan dari mana asal data, kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Berikut ini merupakan *data flow diagram* pada Aplikasi Data Mining AIRPLANE SYSTM :

3.2.2.3.1 DFD Level 1

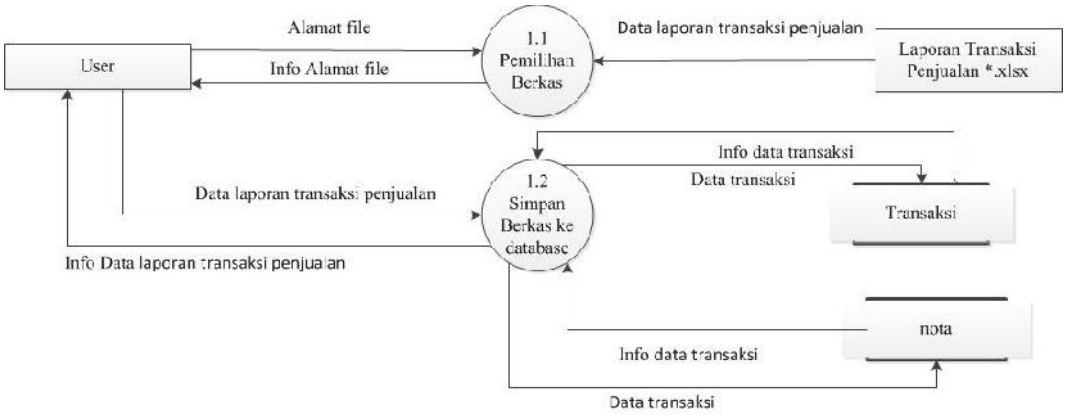
Berikut adalah gambar DFD level 1 dapat dilihat pada Gambar dibawah ini :



Gambar 3.61 DFD Level 1

3.2.2.3.2 DFD Level 2 Proses 1

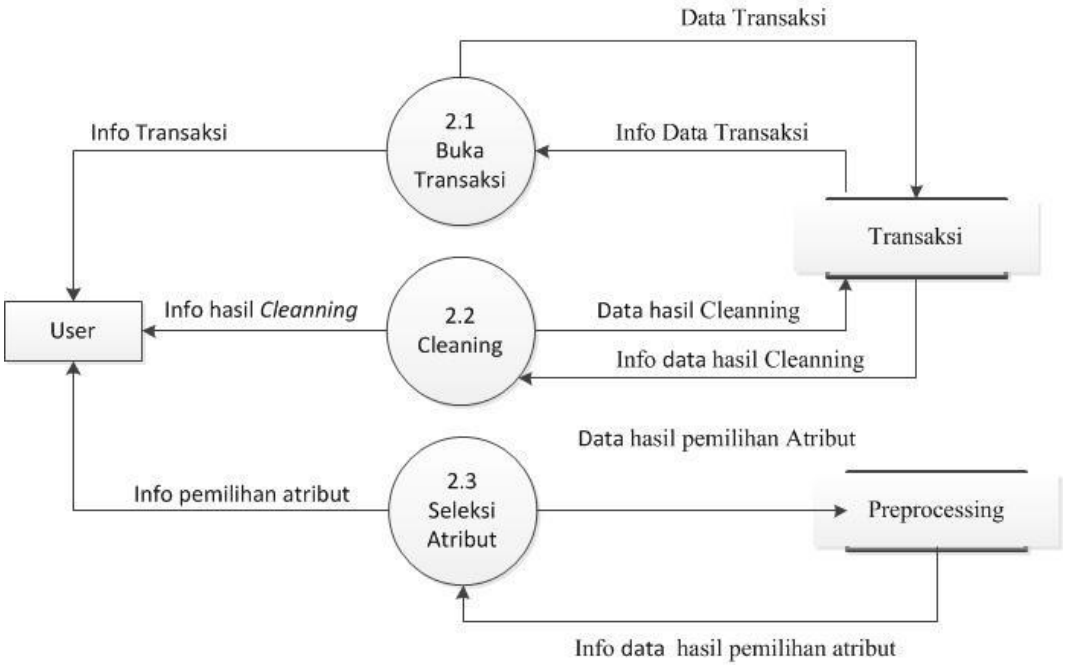
Berikut adalah gambar DFD level 2 Proses 1 dapat dilihat pada Gambar dibawah ini :



Gambar 3.62 DFD Level 2 Proses 1

3.2.2.3.3 DFD Level 2 Proses 2

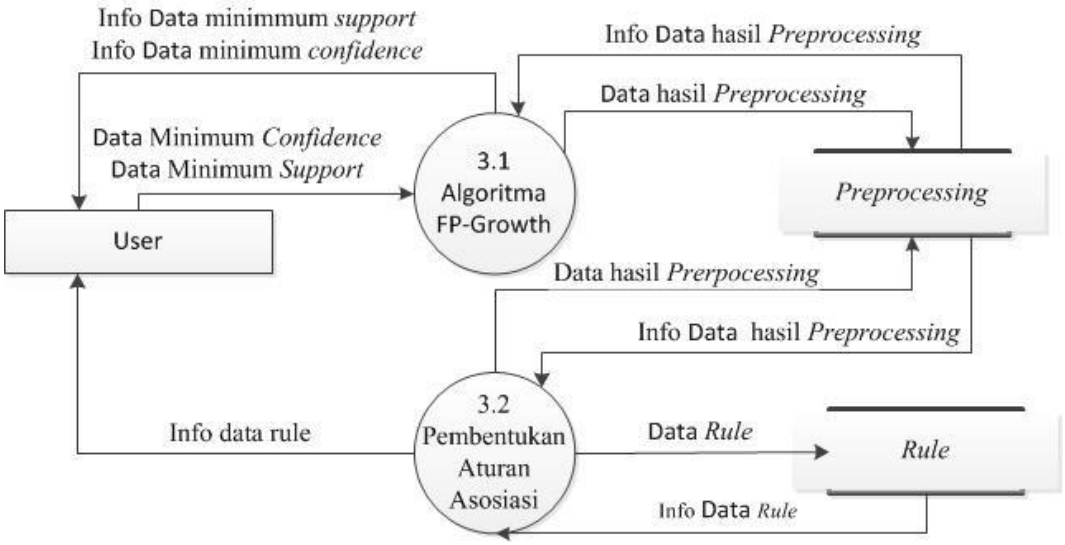
Berikut adalah gambar DFD level 2 Proses 2 dapat dilihat pada Gambar di bawah ini :



Gambar 3.63 DFD Level 2 Proses 2

3.2.2.3.4 DFD Level 2 Proses 3

Berikut adalah gambar DFD level 2 Proses 3 dapat dilihat pada Gambar di bawah ini :



Gambar 3.64 DFD Level 2 Proses 3

3.2.2.4 Spesifikasi Proses

Spesifikasi proses berfungsi untuk menjelaskan proses model aliran yang terdapat pada DFD. Adapun spesifikasi proses dari DFD yang telah dibuat akan dijelaskan pada tabel berikut :

Tabel 3.15 Spesifikasi Proses

No	Proses	Keterangan
1	No.Proses	1
	Nama proses	Pemasukan data transaksi
	Source (sumber)	Pengguna
	Input	Alamat direktori *file* data.xlxs
	Output	Info data alamat file transaksi
	Destination (tujuan)	Simpan data ke *database*
	Logika proses	1. Pengguna memasukan alamat file excel 2. Sistem akan mengekstrak data excel ke dalam database
2	No.Proses	1.1
	Nama proses	Pemilihan Berkas
	Source (sumber)	Pengguna
	Input	Alamat direktori file laporan penjualan
	Output	Memilih file laporan transaksi penjualan .xls yang akan disimpan ke *database*
	Destination(tujuan)	Memilih file laporan tranasksi penjualan untuk dianalisis
	Logika proses	1. Pengguna memasukan file excel 2. Pengguna memilih file transaksi yang akan digunakan untuk dianalisis
3	No.Proses	1.2
	Nama proses	Simpan berkas ke *database*
	Source (sumber)	Data laporan transksi penjualan
	Input	Data laporan transksi penjualan
	Output	Menampilkan data laporan penjualan yang telah disimpan ke dlam *database*
	Destination(tujuan)	Menyimpan data laporan transaksi penjualan ke dalam *database*
	Logika proses	1. Pengguna selesai memilih file 2. Sistem menyimpan data ke dalam tabel data nota dan tabel transaksi
4	No.Proses	2
	Nama proses	Proses *preprocessing* data
	Source (sumber)	Pengguna
	Input	Data transaksi

No	Proses	Keterangan
	Output	Info hasil data transaksi
	Destination(tujuan)	Menghasilkan data hasil *preprocessing* untuk dianalisis
	Logika proses	1. Pengguna selesai memilih file excel 2. File tersimpan di *database* pada tabel transaksi 3. Kemudian sistem akan membersihkan data pada tabel transaksi, pembersihan data berupa penghapusan data transaksi yang memiliki item tunggal , bernilai null dan produk yang bukan dalam jenis baju (tees, t-shirt, ml-ts). 4. Setelah itu, sistem akan memilih atribut No_Nota dan art dari hasil ekstraksi dan menyimpan ke dalam tabel *preprocesing*.
5	No.Proses	2.1
	Nama proses	Buka Transaksi
	Source (sumber)	Data laporan transaksi penjualan
	Input	Data laporan transaksi penjualan
	Output	Menampilkan data transaksi yang ada di *database*
	Destination(tujuan)	Menyimpan data laporan tranasksi penjualan
	Logika proses	1. Pengguna selesai menyimpan file ke database 2. Sistem akan menampilkan hasil data yang telah disimpan.
6	No.Proses	2.2
	Nama proses	Proses *Cleaning*
	Source (sumber)	Data transaksi
	Input	Data transaksi
	Output	Info hasil *cleaning*
	Destination(tujuan)	Membersihkan data transaksi yang memiliki item tunggal, null, serta jenis produk yang tidak digunakan dalam analisis.
	Logika proses	1. Sistem memilih data transaksi yang telah disimpan dalam databas. 2. Sistem menghilangkan data transaksi yang memiliki item tunggal, null serta art yang memiliki jenis (*wallet, jacket, belt, sweater, sw-ml, pants*) 3. Jika tidak memiliki data transaksi yang

No	Proses	Keterangan
		miliki item tunggal maka data hasil *cleaning* disimpan kembali ke *database* transaksi dan dapat digunakan dalam proses selanjutnya
7	No.Proses	2.3
	Nama proses	Proses Seleksi Atribut
	Source (sumber)	Data transaksi
	Input	Data transaksi
	Output	Info hasil seleksi atribut
	Destination(tujuan)	Memilih atribut No_Nota dan art
	Logika proses	1. Sistem men-*scan* data transaksi yang telah disimpan ke dalam *database* 2. Sistem memilih atribut yang akan digunakan dlam proses *data mining*, atribut yang dipilih adala no_nota dan art 3. Hasil seleksi atribut disimpan ke dalam tabel *preprocessing.*
8	No.Proses	3
	Nama proses	Proses Asosiasi
	Source (sumber)	Pengguna
	Input	Data minimum *support*, data minimum *confidence*
	Output	Info Data minimum *support*, data minimum *confidence*
	Destination(tujuan)	Menghasilkan auran asosiasi dengan nilai minimum *support* dan mininum *confidence* yang telah ditentukan pengguna
	Logika proses	1. Pengguna memasukan nilai *minimum support* dan nilai *minimum confidence*, 2. Sistem akan mengurutkan data secara *descending* berdasarkan frekuensi tiap art dari data hasil *preprocessing* 3. Kemudian sistem akan menghapus art yang tidak memenuhi minimum *support* yang telah ditentukan oleh pengguna, 4. Setelah data sudah dihilangkan sesuai dengan masukan dari pengguna sebelumnya, sistem akan men-*scan* data pada no_nota dari setiap transaksi secara berurutan lalu dibuat pohon fp-tree. 5. Kemudian sistem akan melakukan pembentukan *conditional pattern base*,

No	Proses	Keterangan
		conditional fp-tree dan pembentukan *frequent itemset* pada data hasil pembentukan pohon pada langkah 4, 6. Sistem akan membentuk aturan asosiasi dari hasil langkah 5 dan menghitung nilai *confidence* dari tiap-tiap aturan yang dibentuk, 7. Kemudian sistem akan melakukan penghapusan pada data aturan asosiasi yang tidak memenuhi minimum *confidence* 8. Setelah data aturan asosiasi memenuhi aturan minimum confidece, maka sistem akan menghasilkan nilai minimum *support* dan nilai minimum *confidence.*
9	No.Proses	3.1
	Nama proses	Algoritma FP-Growth
	Source (sumber)	Pengguna
	Input	Data nilai minimum *support*, data nilai minimum *confidence*
	Output	Menghasilkan data *itemset* yang *frequent* dan memenuhi minumumm *support*
	Destination(tujuan)	Menghasilkan data *itemset* yang *frequent*
	Logika proses	1. Pengguna memasukan nilai *minimum support* dan nilai *minimum confidence* 2. Jika nilai *minimum support* dan nilai *minimum confidence* kososng, maka sistem akan menampilkan pesan kesalahan. 3. Sistem akan mengurutkan data secara *descending* berdasarkan frekuensi tiap art dari data hasil *preprocessing* 4. Kemudian sistem akan menghapus art yang tidak memenuhi minimum *support* yang telah ditentukan oleh pengguna, 5. Setelah data sudah dihilangkan sesuai dengan masukan dari pengguna sebelumnya, sistem akan men-*scan* data pada no_nota dari setiap transaksi secara berurutan lalu dibuat pohon fp-tree. 6. Kemudian sistem akan melakukan pembentukan *conditional pattern base*, *conditional* fp-tree dan pembentukan

No	Proses	Keterangan
		frequent itemset pada data hasil pembentukan pohon pada langkah 4,
10	No.Proses	3.2
	Nama proses	Pembentukan Aturan Asosiasi
	Source (sumber)	Data *preprocessing*
	Input	Data *preprocessing*
	Output	Menghasilkan data rule yang memenuhi syarat minimum *support* dan minimum *confidence*
	Destination(tujuan)	Menghasilkan aturan asosiasi dengan nilai minimum *support* dan nilai minimum *confidence*
	Logika proses	1. Sistem akan membentuk aturan asosiasi dari hasil pembentukan *frequent itemset* dan menghitung nilai *confidence* dari tiap-tiap aturan yang dibentuk, 2. Kemudian sistem akan melakukan penghapusan pada data aturan asosiasi yang tidak memenuhi minimum *confidence* 3. Setelah data aturan asosiasi memenuhi aturan minimum confidece, maka sistem akan menghasilkan nilai minimum *support* dan nilai minimum *confidence.*

3.2.2.5 Kamus Data

Kamus data berfungsi untuk menjelaskan definisi semua data yang digunakan dan berisi informasi tentang struktur *database* didalam sistem. Berikut adalah kamus data untuk perangkat lunak yang akan dibangun :

Tabel 3.16 Kamus Data

Nama Data	Data Nota
Deskripsi	Memilih *file* data yang akan diproses
Bentuk Data	*File excel* .xlsx
Struktur Data	No Nota+Tanggal

No Nota	{A..Z | 0..9}
Tanggal	{A..Z | a..z | 0..9}
Nama Data	Data Transaksi
Deskripsi	Memilih file data yang akan diproses
Bentuk Data	File excel .xlsx
Struktur Data	No Nota+Tanggal+Brand+Sex+Art+ Colour+XS+ S+ M+ L+ XL+AS+ Qty+ Price+ Amount
No Nota	{A..Z | 0..9}
Tanggal	{A..Z | a..z | 0..9}
Brand	{A..Z | a..z}
Sex	{A..Z | a..z}
Art	{A..Z | a..z}
Colour	{A..Z | a..z}
XS	{0..9}
S	{0..9}
M	{0..9}
L	{0..9}
XL	{0..9}
AS	{0..9}
Qty	{0..9}
Price	{A..Z | a..z}
Amount	{A..Z | a..z}
Nama Data	Data *Preprocessing*
Deskripsi	Data yang digunakan saat pembuatan data *Preprocessing*
Bentuk Data	File excel .xlsx
Struktur Data	No Nota +Art
No Nota	{A..Z | 0..9}
Art	{A..Z | a..z}
Nama Data	Data *Rule*
Deskripsi	Berisi data hasil pembentukan aturan asosiasi

Bentuk Data	Data *Rule*
Struktur Data	*pattern+ support+ Confidence*
pattern	{A..Z | a..z}
support	{0..100}
Confidence	{0..100}

3.2.2.6 Perancangan Sistem

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

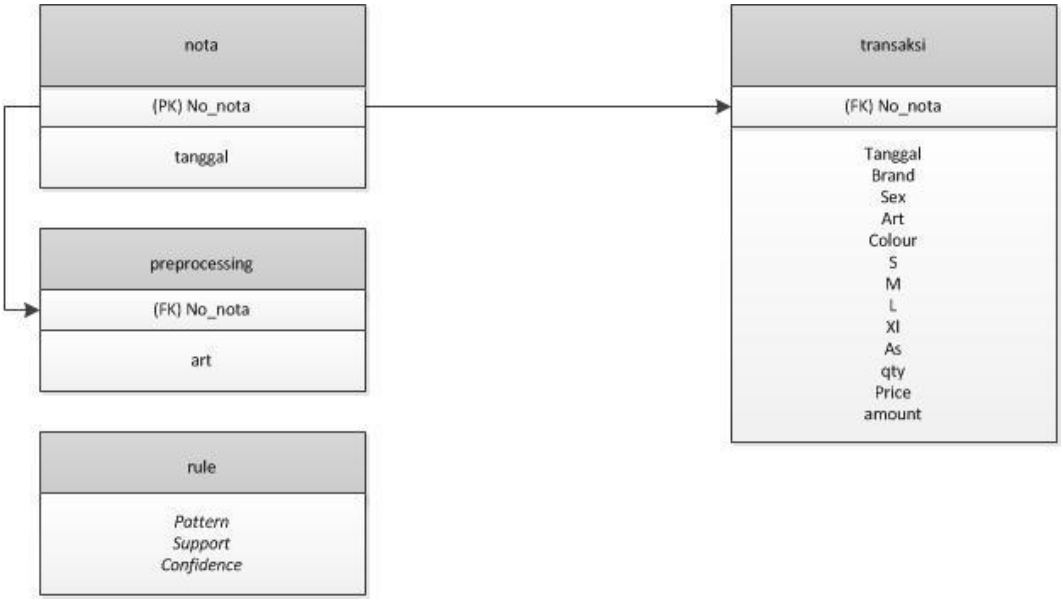
3.2.2.7 Perancangan Basis Data

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

1. Tabel Nota
2. Tabel Transaksi
3. Tabel *Preprocessing*
4. Tabel *Rule*

3.2.2.7.1 Skema Relasi

Skema relasi dari aplikasi yang akan dibangun dapat dilihat pada gambar dibawah ini :



Gambar 3.65 Skema Relasi

3.2.2.7.2 Struktur Tabel

Struktur tabel pada aplikasi yang akan dibangun terdiri dari 4 tabel dengan rincian sebagai berikut :

1. Tabel Nota

Tabel data nota untuk menyimpan data hasil *import* dari file excel.

Tabel 3.17 Data Nota

Nama Atribut	Tipe Data	Panjang	Keterangan
No Nota	Varchar	6	PK
Tanggal	Varchar	30	

2. Tabel transaksi

Tabel data transaksi digunakan untuk menyimpan data detail hasil *import* dari file excel.

Tabel 3.18 Data Transaksi

Nama Atribut	Tipe Data	Panjang	Keterangan
No Nota	Varchar	6	FK references Nota (No Nota)
Tanggal	Varchar	30	
Brand	Varchar	10	
Sex	Varchar	10	
Art	Varchar	20	

Nama Atribut	Tipe Data	Panjang	Keterangan
Colour	Varchar	20	
S	Int	11	
M	Int	11	
L	Int	11	
XL	Int	11	
AS	Int	11	
Qty	Int	11	
Price	Varchar	12	
Amount	Varchar	12	

3. Tabel *preprocessing*

Tabel *preprocessing* digunakan untuk menyimpan data hasil pemilihan data.

Tabel 3.19 *Preprocessing*

Nama Atribut	Tipe Data	Panjang	Keterangan
No Nota	Varchar	6	FK references Nota (No Nota)
Art	Varchar	20	

Karena dibutuhkannya sebuah tempat penyimpanan untuk hasil pencarian *Frequent Itemset*, maka dibuatlah tabel *rule* dengan detail seperti dibawah ini:

1. Tabel *rule*

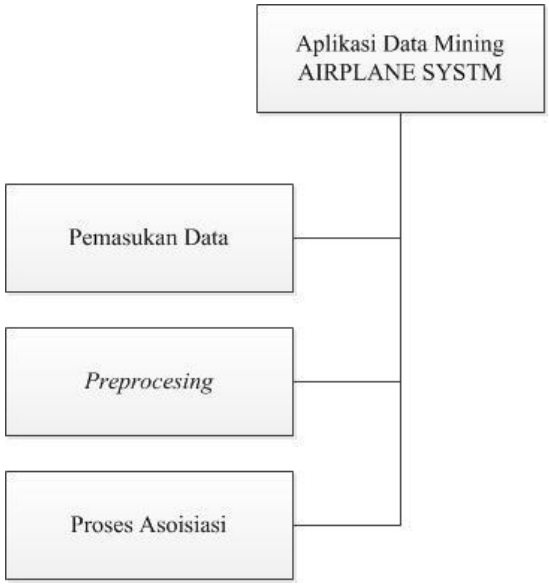
Tabel yang digunakan untuk menyimpan hasil pencarian *Frequent Itemset*.

Tabel 3.20 Proses Asosiasi

Nama Field	Tipe Data	Panjang	Keterangan
rule	Text		
Support	Float		Default Null
Confidence	Float		Default Null

3.2.2.8 Perancangan Struktur Menu

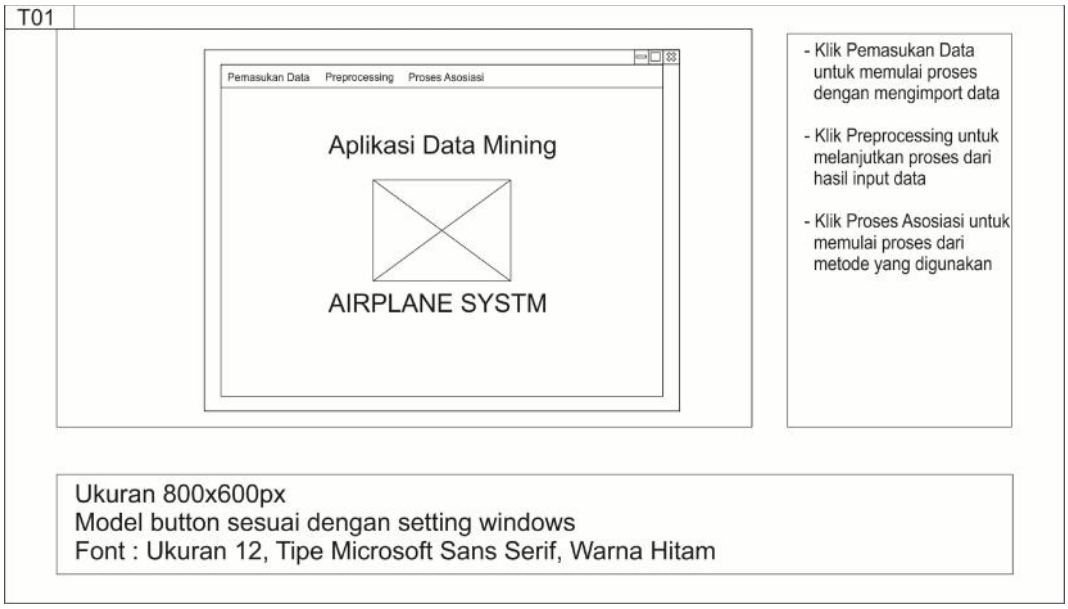
Dalam pembangunan perangkat lunak diperlukan struktur menu untuk memudahkan pengguna dalam menavigasikan fungsional dari sebuah sistem. Adapun perancangna struktur menu untuk aplikasi yang akan dibangun dapat dilihat pada gambar berikut ini :



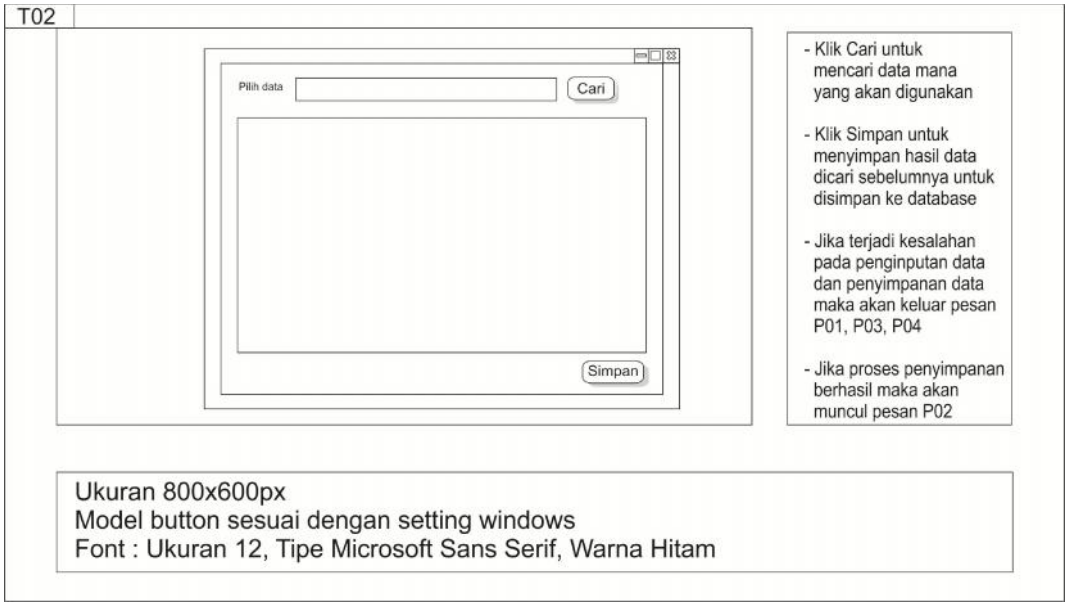
Gambar 3.66 Struktur Menu

3.2.2.9 Perancangan Antar Muka Jendela Aplikasi

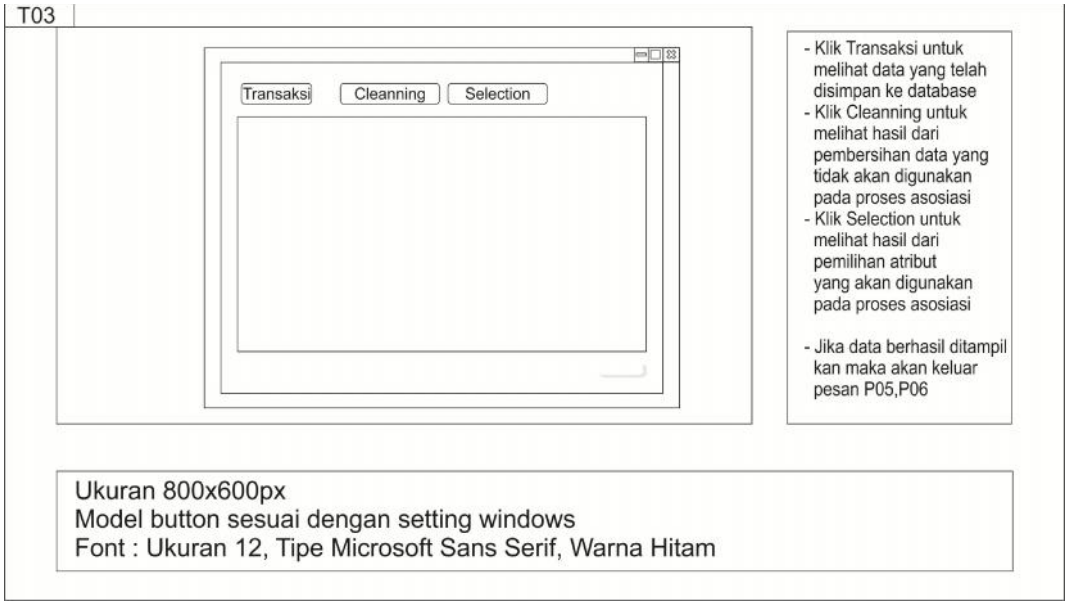
Perancangan antar muka berfungsi untuk memberikan gambaran tentang perangkat lunak yang akan dibangun, supaya lebih memudahkan dalam pembuatan perangkat lunak nantinya. Adapun perancangan antar muka jendela aplikasi yang akan dibangun akan dijelaskan sebagai berikut :



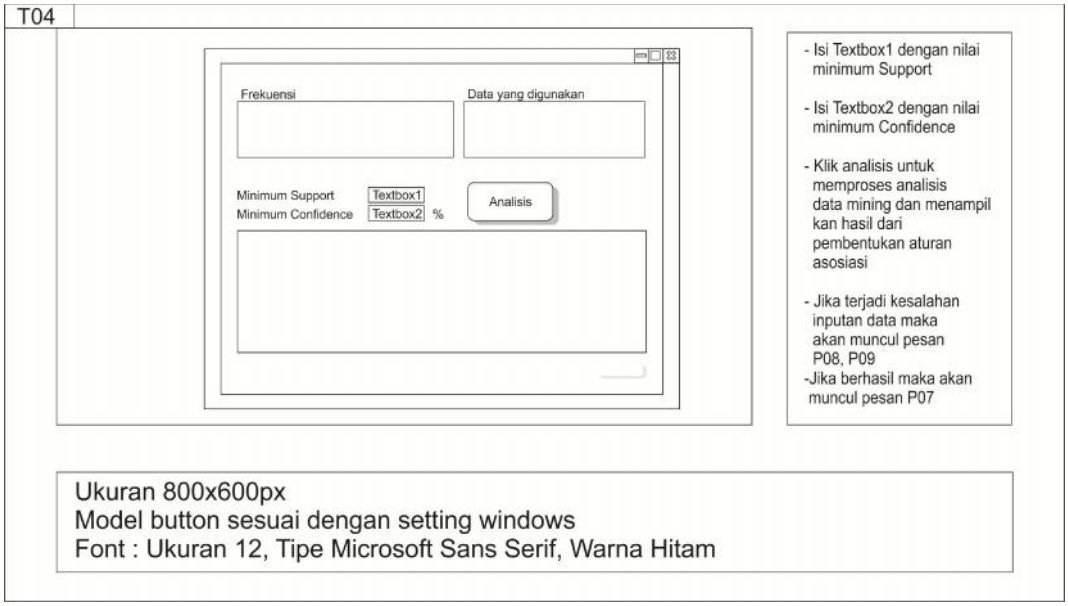
Gambar 3.67 Rancangan Antar Muka Halaman Utama



Gambar 3.68 Rancangan Antar Muka Halaman Pemasukan Data



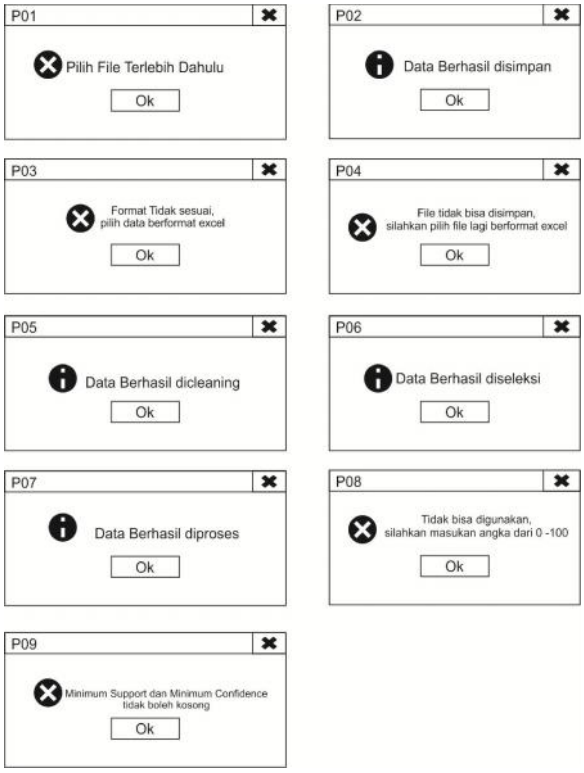
Gambar 3.69 Rancangan Antar Muka Halaman *Preprocessing* Data



Gambar 3.70 Rancangan Antar Muka Halaman Proses Asosiasi

3.2.2.10 Perancangan Antar Muka Pesan

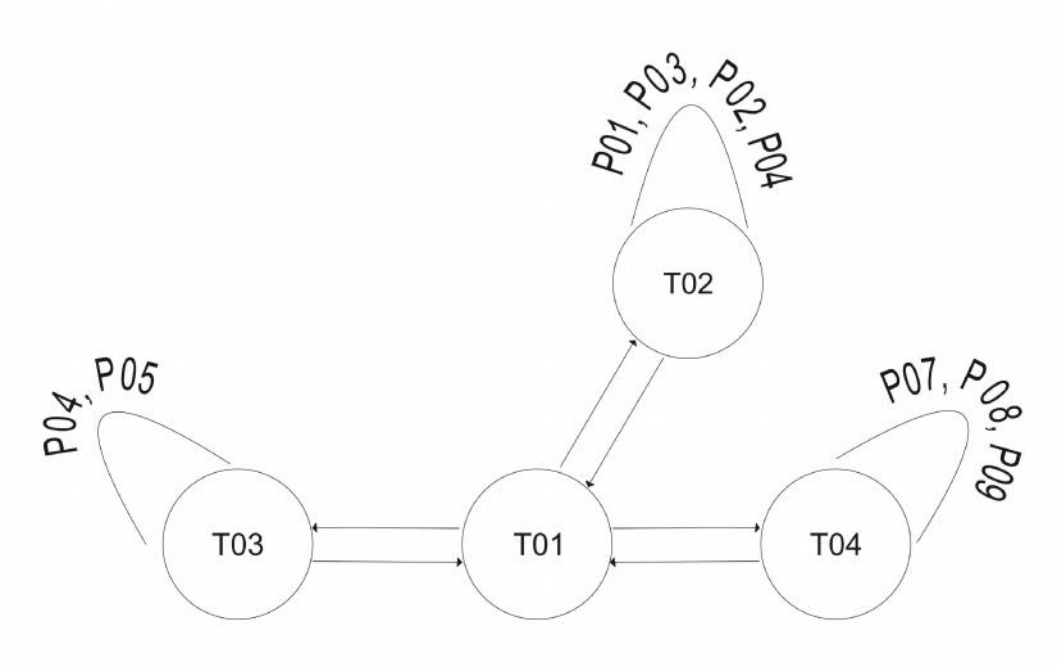
Adapun perancangan pesan dapat dilihat pada gambar dibawah ini :



Gambar 3.71 Perancangan Antar Muka Pesan

3.2.2.11 Perancangan Semantik

Perancangan semantik dapat dilihat pada gambar dibawah ini :



Gambar 3.72 Jaringan Semantik