

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis sistem

Proses menentukan produk terunggul CV.Anugrah Jaya Muliya dilakukan dengan cara menganalisa pola data transaksi penjualan yang didapat dari setiap bulanya, dengan menganalisa sebuah pola yang terkandung didalam data transaksi CV.Anugrah Jaya Muliya mendapatkan informasi mengenai produk-produk terunggul atau produk yang paling diminati para konsumennya, dimana hasil informasi tersebut digunakan sebagai acuan untuk tindakan dalam mengembangkan bisnis. Seperti, memperbanyak jumlah persediaan produk, dan mengembangkan produk terunggul di setiap bulanya. Adapun atribut dalam data transaksi yang dianalisa adalah tanggal transaksi, nama konsumen, dan nama produk, dari ketiga atribut tersebut pihak CV.Anugrah Jaya Muliya menganalisa pola data transaksi, misalnya dari pola data transaksi satu bulan diketahuinya beberapa konsumen yang berbeda telah membeli produk atau item yang sama, dari diketahuinya informasi tersebut dilakukan pencatatan terhadap jumlah produk atau item yang paling sering dibeli konsumen dari setiap bulanya.

Penggunaan alat bantu yang terbatas dan lamanya dalam penentuan produk terunggul CV.Anugrah Jaya Muliya menghambat proses penentuan. Sebab semakin banyak data transaksi penjualan dalam kurun waktu tertentu semakin lama juga proses menentukannya, karena proses yang dilakukan masih dilakukan dengan pencatatan, sehingga hal ini membuat pihak perusahaan kesulitan untuk menentukan produk mana yang paling diminati oleh konsumen.

3.2 Hasil analisis

Dari tahap analisis dapat diketahui kendala yang terjadi. Sistem yang akan dibangun nantinya dapat mempermudah proses menentukan produk terunggul CV.Anugrah Jaya Muliya, serta mampu memberikan rekomendasi

keputusan yang terbaik dalam menentukan produk-produk yang seharusnya dikembangkan. Dimana dalam prosesnya nanti terdapat satu entitas yaitu bagian admin CV.AJM umum yang bertugas sebagai entitas aktif yang bertanggung jawab terhadap pengolahan data transaksi Untuk dapat melakukan proses Data mining *association rule*, maka perlu ditentukan beberapa variabel penilaian sebagai dasar perhitungan. Variabel yang digunakan meliputi 2 nilai parameter yaitu nilai minimum *support* dan minimum *confidence*.

- a. Nilai minimum *support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item dari keseluruhan total transaksi, nilai *support* ditentukan antara 0,0 sampai 1,0. Nilai 0,0 diambil dari perhitungan presentas paling kecil yaitu $0\%(0/100=0)$.

Nilai 1,0 diambil dari perhitungan presentase paling besar yaitu $100\%(100/100=1)$.

- b. Nilai *confidence* adalah suatu ukuran yang menunjukkan kuatnya hubungan antar item dalam *association rule/pola*, nilai *confidence* ditentukan antara 0,0 sampai 1,0

Nilai 0,0 diambil dari perhitungan presentase paling kecil yaitu $0\%(0/100=0)$

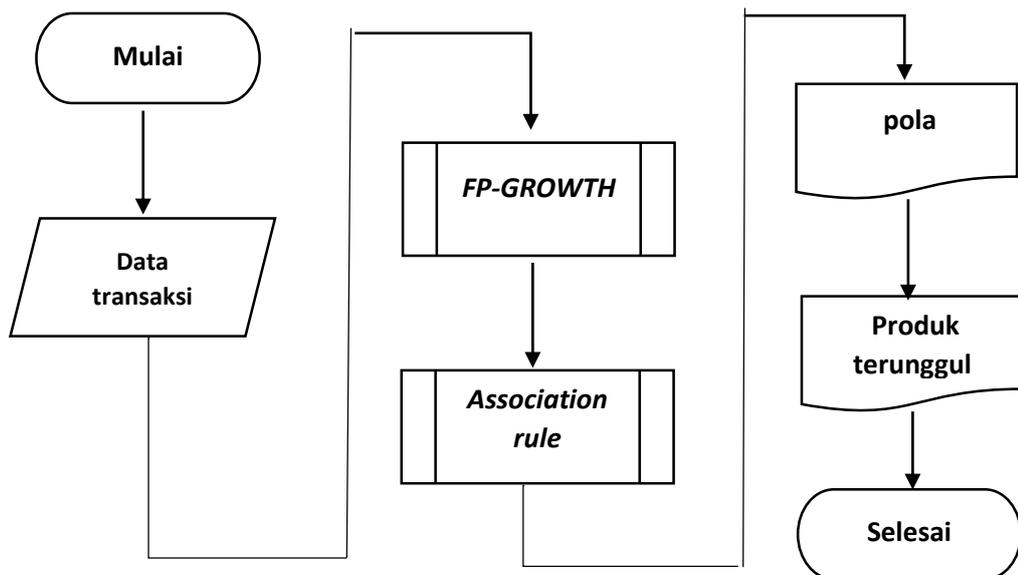
Nilai 1,0 diambil dari perhitungan presentase paling besar yaitu $100\%(100/100=1)$.

Metode *FP-Growth (Frequent Pattern Growth)* secara garis besar merupakan proses menentukan himpunan data yang paling sering muncul (*frequent itemset*) dari kumpulan data, yang pertama sebelum ke proses *FP-Growth*, *FP-Growth* menggunakan struktur tree yang disebut *FP-Tree*, Proses penyusunan *FP-Tree* dari mulai representasi awal transaksi untuk menentukan *frequent itemset*, setelah didapaknya *frequent itemset* kemudian dilanjutkan ke perhitungan *association rule*, menghitung rule yang dihasilkan dari penerapan metode *FP-Growth* untuk mendapatkan pola terbaik dari pola-pola yang ada. Hasil akhir pola dari perhitungan *association*

rule merupakan hasil rekomendasi sebagai bahan acuan terbaik dalam menentukan item atau produk yang paling sering dibeli oleh konsumen.

3.2.1 Gambaran umum sistem yang dibangun

Dalam gambaran umum ini digambarkan algoritma secara umum untuk semua proses yang ada dalam Sistem menentukan produk terunggul dari setiap bulanya. Proses diawali dengan memasukkan data transaksi penjualan, dari data transaksi penjualan atribut yang digunakan hanya dua atribut yaitu tanggal transaksi dan nama produk, kemudian admin memasukkan nilai minimum *support* dan *confidence* dan dilakukan proses perhitungan *association rule* untuk proses menentukan *frequent itemset*. Berikut gambaran dari proses sistem yang dibangun dapat dilihat pada **Gambar 3.1**



Gambar 3.1 Gambaran umum sistem yang dibangun

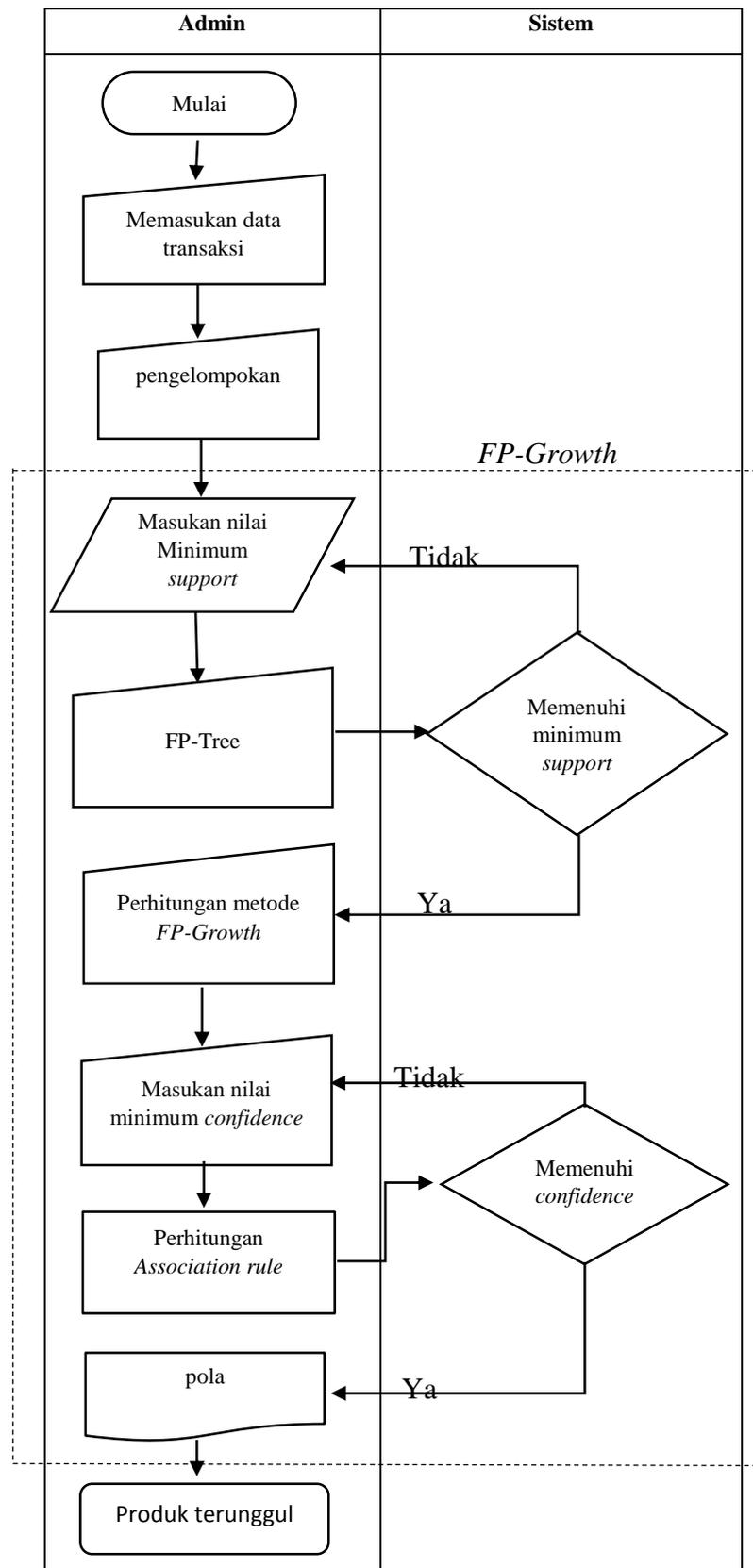
3.2.2 Detail Diagram Alir Sistem *FP-Growth*

Diagram alir ini berfungsi untuk menggambarkan alur algoritma serta proses yang digunakan pada sistem menentukan *frequent itemset* atau pola kebiasaan konsumen dalam belanja dengan menggunakan metode *FP-Growth* yang berfungsi untuk menghitung dan menentukan himpunan data yang

paling sering muncul (*frequent itemset*). Berikut adalah detail diagram alir sistem *FP-Growth* pada **Gambar 3.2**

Keterangan umum proses yang terjadi pada **Gambar 3.2**

1. Admin memasukkan data transaksi yang akan di proses untuk penyeleksian atribut yang akan digunakan ke proses data mining.
2. Pengelompokan data yang menjadi fokus penelitian yang meliputi tanggal transaksi dan nama barang yang dibeli.
3. admin memasukkan batasan nilai minimum *support* nilai yang akan dijadikan acuan dalam menentukan *item-item* yang memenuhi minimum *support* dan akan di proses ke proses *FP-Tree*.
4. proses menentukan jumlah frekuensi item dari minum *support* yang ditentukan.
5. Melakukan proses perhitungan dengan metode *FP-Growth*, yaitu menentukan dua item yang memenuhi minimum *support*.
6. admin memasukan batasan nilai minimum *confidence*, nilai yang akan dijadikan sebagai acuan dalam menentukan dua sampai tiga item yang memenuhi minimum *confidence*.
7. Melakukan proses perhitungan *association rule*, untuk mendapatkan rule atau pola yang memenuhi nilai minimum *confidence*.



Gambar 3.2 Detail Diagram Alir Sistem *FP-Growth*

3.2.3 Kebutuhan Pengguna Sistem

Aplikasi ini mempunyai 1 jenis pengguna, yaitu :

1) Admin (CV.AJM)

Dalam fungsinya admin selaku pemegang hak akses utama dalam sistem ini. Adapun kebutuhan fungsional admin tersebut adalah sebagai berikut :

- a. Admin dapat melakukan akses pada menu master data, dimana pada menu tersebut memiliki sub menu diantaranya:
 - i. Petunjuk : admin dapat melihat prosedur cara menggunakan sistem untuk melakukan proses penentuan pola kebiasaan konsumen dalam berbelanja.
 - ii. Data transaksi : admin dapat mengubah, menambah dan menghapus data transaksi.
 - iii. *FP-Growth* : admin dapat melihat, menentukan tanggal transaksi, dan memasukan nilai minimum *support*, minimum *confidence*.

3.2.4 Identifikasi Kebutuhan Sistem

Aplikasi *FP-Growth* untuk menentukan *frequent itemset* ini memiliki 4 tampilan utama yaitu, menu petunjuk, menu input data transaksi, data transaksi, menu *FP-Growth* yang meliputi sub sub proses penentuan *frequent itemset*.

1. Menu utama

Pada menu utama aplikasi ini memiliki 4 fitur menu yaitu, petunjuk, input data transaksi, data transaksi dan proses *FP-Growth* , dimana masing-masing memiliki sub menu tersendiri.

2. Menu petunjuk

Menu petunjuk berfungsi untuk memberikan informasi cara menggunakan sistem untuk melakukan proses penentuan pola kebiasaan konsumen dalam berbelanja.

3. Menu data transaksi

Menu data transaksi sendiri berfungsi untuk menambahkan, merubah dan menghapus data transaksi.

4. Menu proses *FP-Growth*

Menu ini merupakan tampilan untuk memproses data transaksi dan dapat mengetahui hasil akhir perhitungan sistem dimana akan muncul pengelompokan data, frekuensi item, pengurutan data berdasarkan frekuensi dan hasil pola yang akan dijadikan pedoman untuk menentukan produk terunggul di setiap bulanya.

3.3 Representasi Model

Data yang akan dijadikan perhitungan dalam sistem *FP-Growth* dalam menentukan pola data transaksi penjualan akan melalui beberapa tahap sesuai dengan rule yang ada. Penilaian yang digunakan dalam sistem yang dibuat ini menggunakan metode *FP-Growth* (*Frequent Pattern Growth*) dengan memakai bantuan struktur *tree* yang disebut *FP-Tree* dan selanjutnya perhitungan *association rule* untuk menjamin kuatnya nilai dari pola yang dihasilkan.

3.3.1 Representasi Data, dan langkah-langkah Data Mining penyelesaian menghitung frekuensi kemunculan itemset menggunakan metode *FP-Growth*

A.1 Pengelompokan data

Sumber data utama yang digunakan dalam penelitian ini berasal dari 15 data transaksi bulan januari 2014 di CV.AJM. Data tersebut nantinya akan diolah untuk mendapatkan pengetahuan berupa pola pembelian *item* oleh konsumen yang bisa digunakan sebagai acuan dalam menentukan produk terunggul di setiap bulanya.

A.2 Penyeleksian data

Pada data transaksi penjualan selama 1 bulan terdapat beraneka ragam *item* yang dibeli oleh konsumen. Dalam penelitian ini, penulis mengambil 15 produk *item* sebagai bahan penelitian. Sehingga pada tahap penyeleksian data

ini akan mengambil data transaksi yang mengandung atribut dan *item* yang sudah ditentukan oleh penulis dan transaksi yang tidak mengandung atribut dan item yang diteliti akan dibuang. Adapun struktur data penjualan *itemset* yang akan diteliti dapat dilihat pada tabel 3.1

Tabel 3.1 Struktur Data Penjualan CV.AJM

DOKUMEN	KETERANGAN		
Detail Data Penjualan	Deskripsi	Data ini berisi transaksi penjualan yang terjadi di CV.AJM	
	Fungsi	Untuk mengetahui produk yang dibeli oleh konsumen	
	Format	Microsoft Excel (.xlsx)	
	Atribut	Tanggal transaksi	Tanggal penjualan
		Nama	Nama konsumen
		Alamat	Alamat konsumen
		Kota	Kota konsumen
		NoHp	No handphone konsumen
		Id produk	Id produk
		Nama produk	Nama produk yang dibeli
		Harga produk	Harga dari produk yang dibeli

A.3 Preprocessing/ Cleaning

Proses ini dilakukan untuk membersihkan data yang tidak relevan berdasarkan aturan yang diterapkan dalam memenuhi tujuan proses *mining*. Mencari pola pembelian serta memberi informasi tentang pemilihan produk terunggul di setiap bulanya. Maka aturan yang digunakan jika dalam satu transaksi hanya terdapat satu *item* produk maka transaksi tersebut akan dibersihkan dari data, dan data penjualan barang yang akan disimpan untuk diproses ke tahap selanjutnya yaitu tanggal transaksi dan nama item atau nama produk saja, sehingga bentuk data transaksi penjualan untuk analisa akan menjadi seperti yang terlihat pada tabel 3.2 :

Tabel 3.2 Data penjualan produk untuk Analisa

Tanggal transaksi	Nama produk
04/01/2014	Donna
	Berete
04/01/2014	Anthra cuba
	Esther
	Berete
08/01/2014	Febby series
	Water fall
	Esther
	Alexander
11/01/2014	Febby series
	Charisa dress
11/01/2014	Donna
	Camara
13/01/2014	Flow MD
	Alexander
15/01/2014	Dior MD
	Berete
15/01/2014	Saffa
	Alexsander
	Esther
	Calista
18/01/2014	Alexsander
	Esther
19/01/2014	Adela Top
	Berete
21/01/2014	Donna
	Berete
	Esther

A.4 Transformasi data

Tahap transformasi data perlu dilakukan, karena Dalam tahap pencarian *frequent itemset* algoritma yang akan digunakan adalah *Frequent pattern Growth*. Algoritma ini memiliki dasar pengetahuan mengenai *frequent itemset* yang telah diketahui sebelumnya untuk memproses informasi lebih lanjut. Dalam algoritma *Fp-growth* terdapat tiga tahapan penting, namun sebelum masuk pada tahapan untuk mempermudah pembentukan *frequent*

itemset akan dilakukan transformasi data terhadap produk yang terdapat pada data yang sudah siap digunakan untuk proses *mining*, proses transformasi ini memberikan inisial pada setiap produk yang terdapat pada transaksi, Proses transformasinya adalah nama produk diberi inisial menjadi huruf dari A, B, C s/d jumlah item yang ditentukan. Sehingga contoh proses transformasinya seperti nama barang Adela top \rightarrow Aa, dan Anta cuba \rightarrow Ba, untuk memperjelas dapat dilihat pada tabel 3.3

Tabel 3.3 Hasil Data transformasi analisa produk

no	Nama item	Inisial item
1	domna	Fa
2	berete	Da
3	antha cuba	Ba
4	esther	Ca
5	febby series	Ia
6	water fall	Oa
7	alexsander	Ea
8	charisa dress	Ja
9	camara	Ha
10	flow MD	Ma
11	dior MD	Ka
12	saffa	Ga
13	calista	Na
14	adela top	Aa
15	martha	La

Setelah dilakukan transformasi terhadap setiap produk maka selanjutnya masuk pada algoritma *Fp-growth*. Tahapan-tahapannya adalah sebagai berikut :

B.1 Penyelesaian dalam Tahap *FP-Tree*

Pada tahap ini, *FP-Tree* dibuat untuk digunakan bersamaan dengan algoritma *FP-Growth* untuk menentukan *frequent itemsets*, Penulis mengambil contoh data sebanyak 15 transaksi yang sudah disederhanakan seperti terlihat pada tabel 3.3. Penulis memberikan batasan nilai minimum

support 0.2 atau sama dengan 20%. dan *confidence* 0.75 atau sama dengan 75%, Dalam membangun *FP-Tree* diperlukan dua kali penelusuran *database*.

Tabel 3.4 Data transaksi yang telah di kelompokkan

TID	item
1	Fa, Da
2	Ba, Ca, Da
3	Ia, Oa, Ca, Ea
4	Ia, Ja
5	Fa, Ha
6	Ma, Ea
7	Ka, Da
8	Ga, Ea, Ca, Na
9	Ea, Ca
10	Aa, Da
11	Fa, Da, Ca
12	Da
13	La, Ba, Ca
14	Oa, Ea, Ba, Ca
15	Ia, Oa, Ka

Penelusuran *database* yang pertama digunakan untuk menghitung nilai *support* masing-masing *item* dan memilih *item* yang memenuhi nilai minimum *support*. Hasil dari proses penelusuran *database* yang pertama adalah diketahuinya jumlah frekuensi kemunculan tiap *item* yang ada didalam *database*. Hasilnya dapat dilihat pada tabel 3.4

Tabel 3.5 Frekuensi kemunculan tiap item

item	Frekuensi
Aa	1
Ba	3
Ca	7
Da	6
Ea	5
Fa	3
Ga	1
Ha	1
Ia	3
Ja	1
Ka	2
La	1
Ma	1
Na	1
Oa	3

Dari hasil tabel 3.4 diperoleh *Itemset* yang memiliki frekuensi di atas minimum *support count* ≥ 3 yang didapat dari perhitungan $\text{minsupport} \times \text{jumlah transaksi}$ $0,2 \times 15 = 3$ yaitu Ba,Ca,Da,Ea,Fa,Ia dan Oa yang kemudian diberi nama *Frequent List* . dan mengurutkannya berdasarkan dari jumlah frekuensi kemunculan terbesar yang seperti terlihat pada tabel 3.5. Sementara *item* Aa,Ga,,Ha,Ja,Ka,La,Ma, dan Na dibuang karena tidak memenuhi *minimum support*.

Tabel 3.6 Frequent list

item	Frekuensi
Ca	7
Da	6
Ea	5
Ba	3
Fa	3
Ia	3
Oa	3

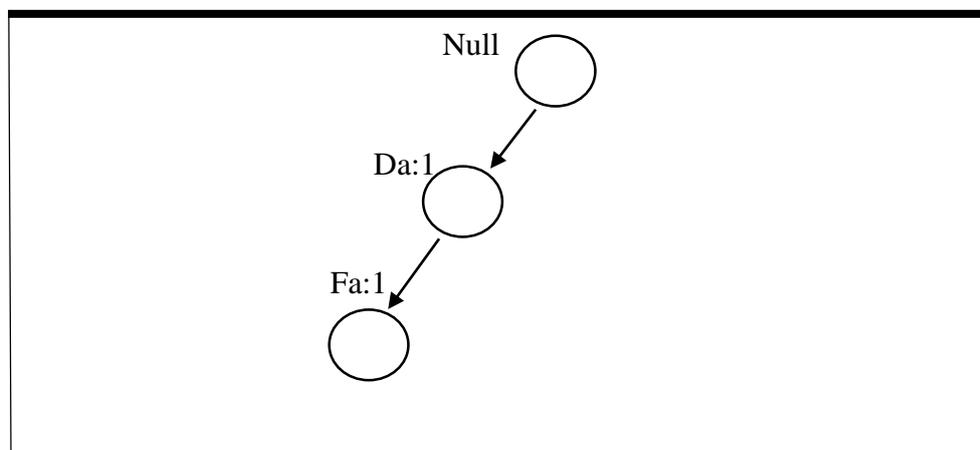
Setelah diperoleh *Frequent List*, kembali ke data transaksi dan urutkan *Itemset* pada tiap transaksi berdasarkan dari frekuensi paling tinggi seperti

pada tabel 3.6, Kemudian membuat *tree* berurut berdasarkan transaksi ID nya.

Tabel 3.7 Transaksi yang telah teurut berdasarkan *frequent list*

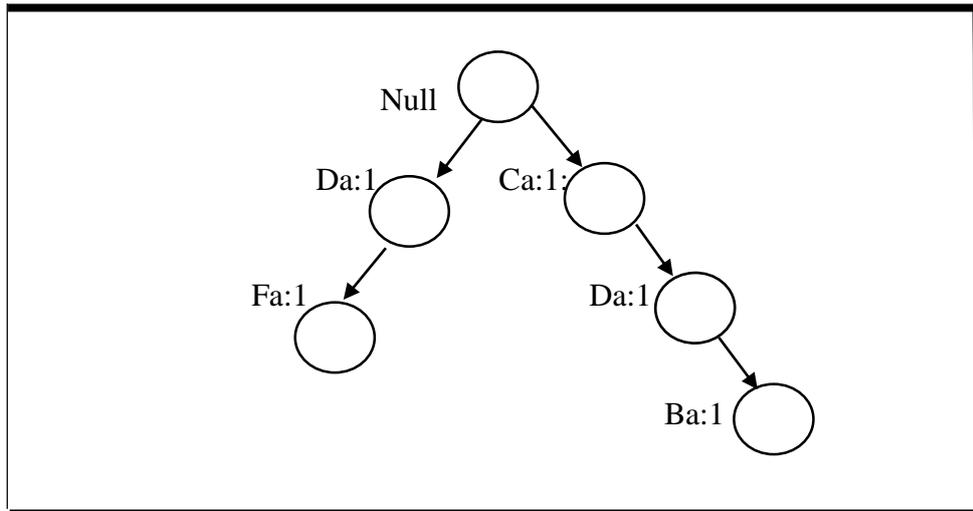
TID	item
1	Da,Fa
2	Ca,Da,Ba
3	Ca,Ea,Ia,Oa
4	Ia,
5	Fa,
6	Ea,
7	Da,
8	Ca,Ea
9	Ca,Ea
10	Da,
11	Ca,Da,Fa
12	Da
13	Ca,Ba
14	Ca,Ea,Ba,Oa
15	Ia,Oa

Setelah *item* disusun ulang berdasarkan *F-list*, dilakukan penelusuran *database* yang kedua yaitu membaca tiap transaksi diawali dengan membaca TID 1 untuk membuat *FP-Tree*. TID 1 {Da,Fa} akan membuat simpul Da dan Fa sehingga terbentuk lintasan $\{\} \rightarrow Da \rightarrow Fa$ dengan *support count* awal bernilai satu.



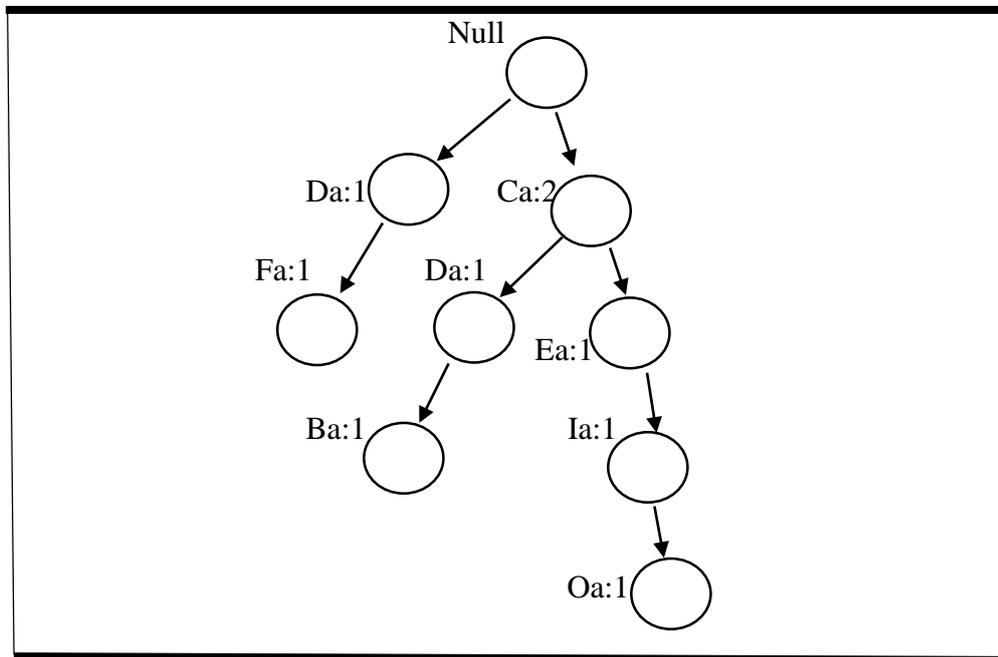
Gambar 3.3 Hasil pembacaan TID 1

Setelah pembacaan TID 1, maka selanjutnya membaca TID 2 yaitu {Ca, Da, Ba} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Da \rightarrow Ba$ dengan *support count* awal bernilai satu juga. Walaupun Da ada pada TID 1, tapi karena *prefix* transaksinya tidak sama dengan lintasan sebelumnya, maka TID 2 membuat lintasan baru.



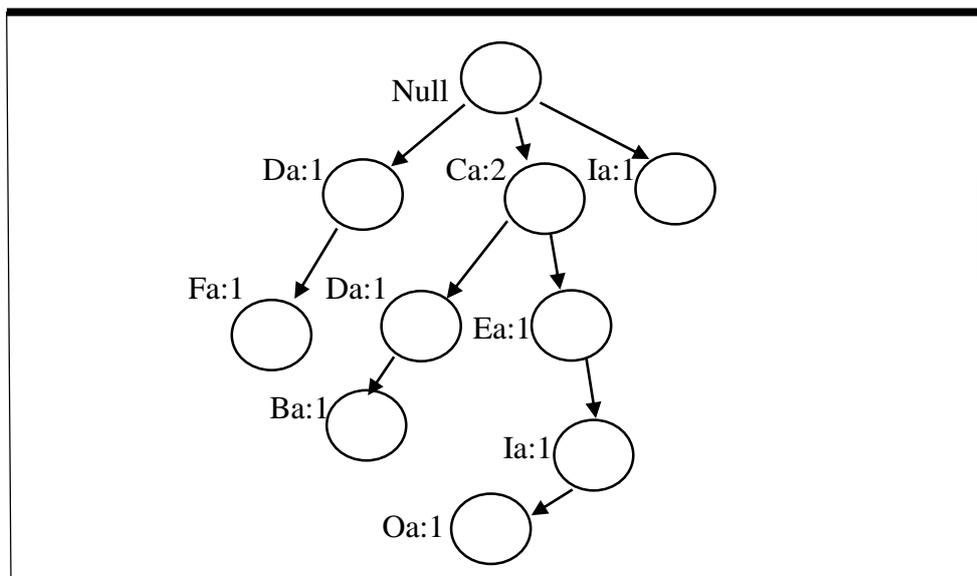
Gambar 3.4 Hasil pembacaan TID 2

Setelah pembacaan TID 2, maka selanjutnya membaca TID 3 yaitu {Ca, Ea, Ia, Oa} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Ea \rightarrow Ia \rightarrow Oa$. Karena memiliki *prefix* yang sama dengan lintasan TID 2 yaitu item Ca, maka lintasan TID 3 bisa ditimpahkan pada lintasan TID 2. Setelah itu tambahkan *support count* item Ca 1 menjadi 2 karena telah dilewati sebanyak dua kali, sedangkan Ia dan Oa membuat lintasan sendiri dari item Ca masing-masing bernilai *support count* Satu.



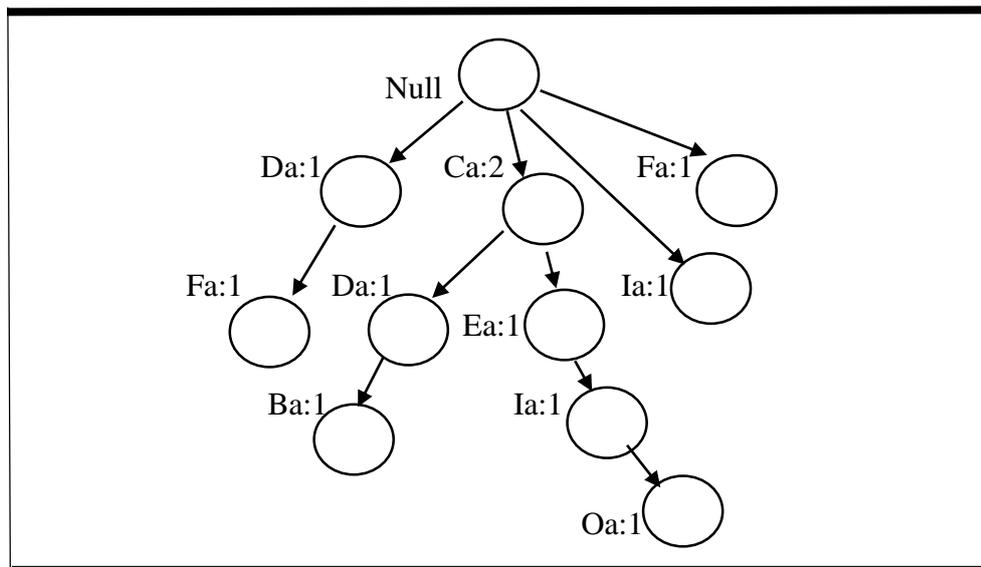
Gambar 3.5 Hasil pembacaan TID 3

Setelah pembacaan TID 3, maka selanjutnya membaca TID 4 yaitu {Ia} sehingga membentuk lintasan $\{\} \rightarrow Ia$. Karena tidak memiliki *prefix* yang sama maka lintasan TID 4 membuat lintasan baru dengan nilai *support count* satu



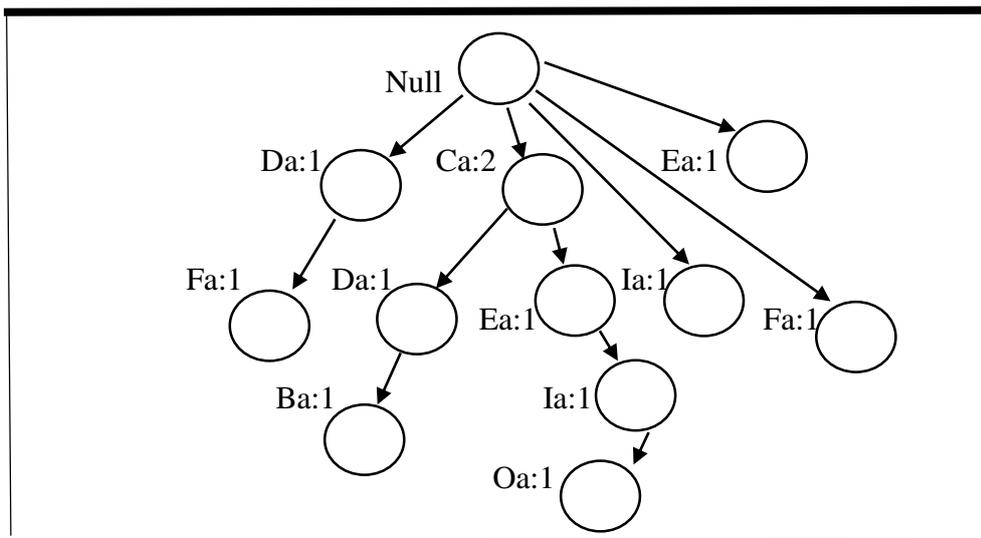
Gambar 3.6 Hasil pembacaan TID 4

Setelah pembacaan TID 4, maka selanjutnya membaca TID 5 yaitu {Fa} sehingga membentuk lintasan $\{\} \rightarrow Fa$. Karena *prefix* transaksinya tidak ada yang sama dengan lintasan sebelumnya, maka TID 5 dibuat lintasan baru dengan nilai *support count* satu.



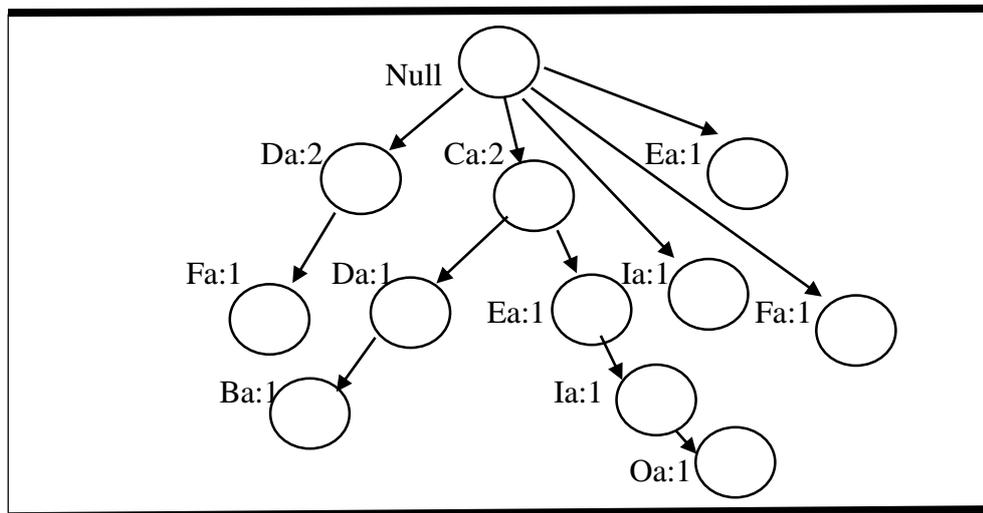
Gambar 3.7 Hasil pembacaan TID 5

Setelah pembacaan TID 5, selanjutnya membaca TID 6 yaitu {Ea} sehingga membentuk lintasan $\{\} \rightarrow Ea$. Karena *prefix* transaksinya tidak ada sama dengan lintasan sebelumnya, maka TID 6 dibuat lintasan baru.



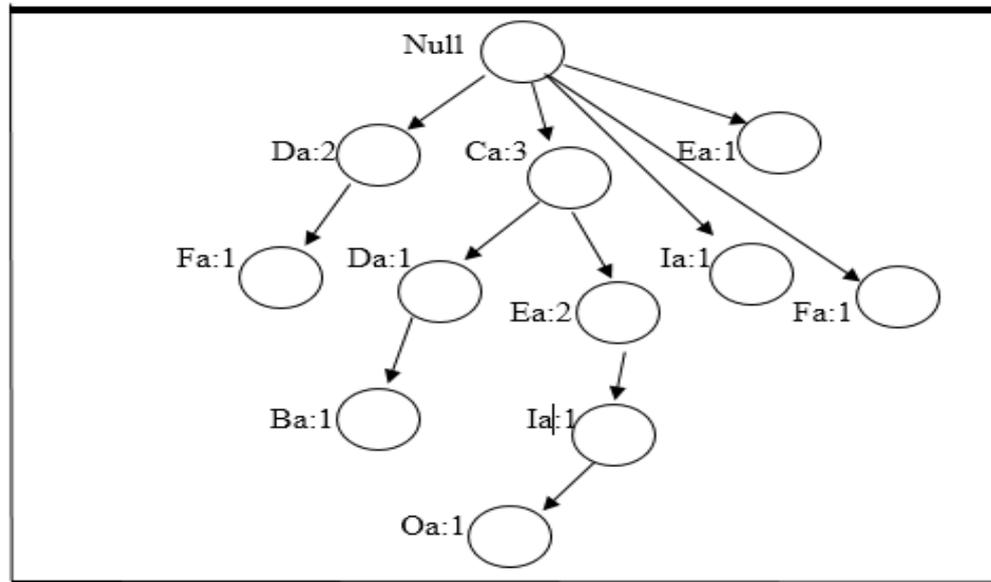
Gambar 3.8 Hasil pembacaan TID 6

Selanjutnya membaca TID 7 yaitu {Da} sehingga membentuk lintasan $\{\} \rightarrow Da$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya yaitu Da, maka TID 6 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count nya.



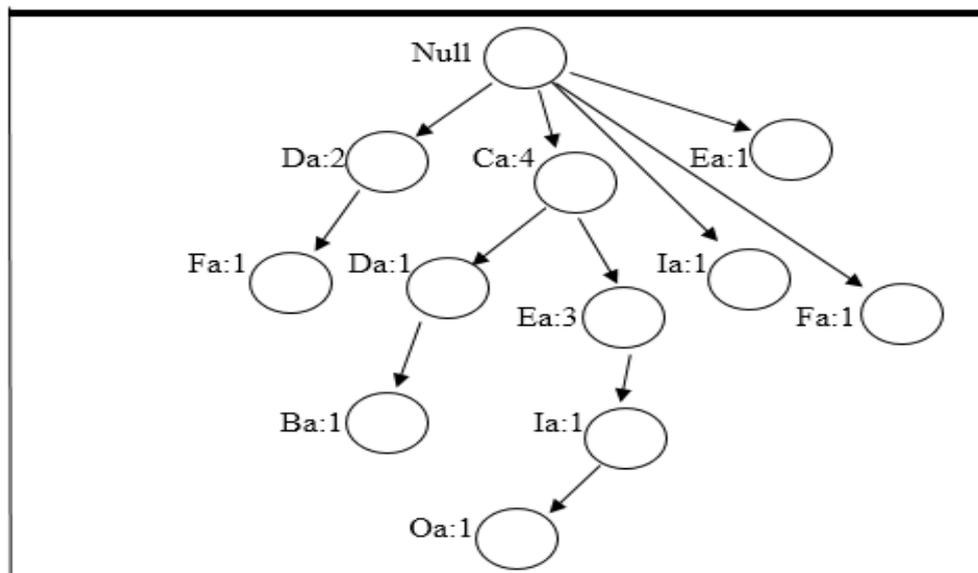
Gambar 3.9 Hasil pembacaan TID 7

Setelah pembacaan TID 7, maka selanjutnya membaca TID 8 yaitu {Ca,Ea,} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Ea$. Karena memiliki *prefix* yang sama dengan lintasan sebelumnya yaitu Ca,Ea maka lintasan TID 8 bisa ditimpahkan pada lintasan sebelumnya. Setelah itu tambahkan *support count* item Ca yang sebelumnya 2 menjadi 3 dan Ea 1 menjadi 2 karena Ca telah dilewati sebanyak tiga kali dan Ea dua kali.



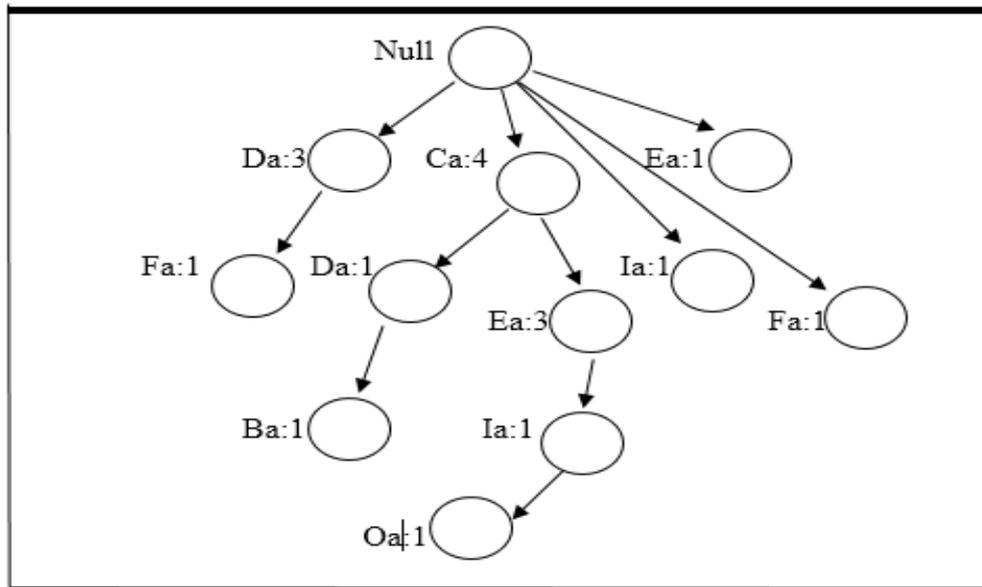
Gambar 3.10 Hasil pembacaan TID 8

Setelah pembacaan TID 8, maka selanjutnya membaca TID 9 yaitu {Ca,Ea,} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Ea$. Karena memiliki *prefix* yang sama dengan lintasan sebelumnya yaitu item Ca,Ea maka lintasan TID 9 bisa ditimpahkan pada lintasan sebelumnya. Setelah itu tambahkan *support count* item Ca yang sebelumnya 3 menjadi 4 dan Ea 2 menjadi 3 karena Ca telah dilewati sebanyak empat kali dan Ea tiga kali.



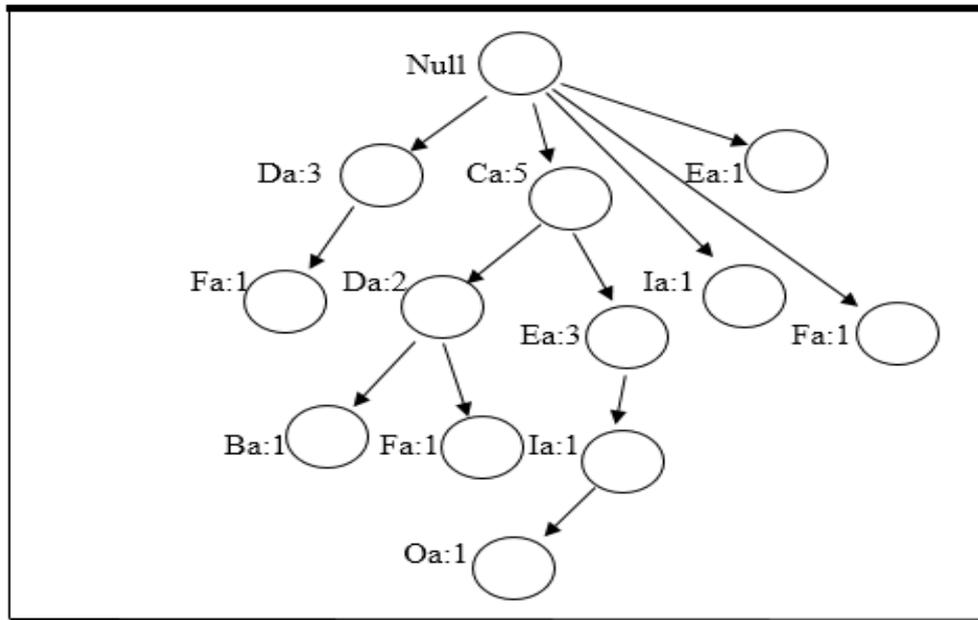
Gambar 3.11 Hasil pembacaan TID 9

Setelah pembacaan TID 9, maka selanjutnya membaca TID 10 yaitu {Da} sehingga membentuk lintasan $\{\} \rightarrow Da$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya, maka TID 10 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count lintasan Da menjadi 3



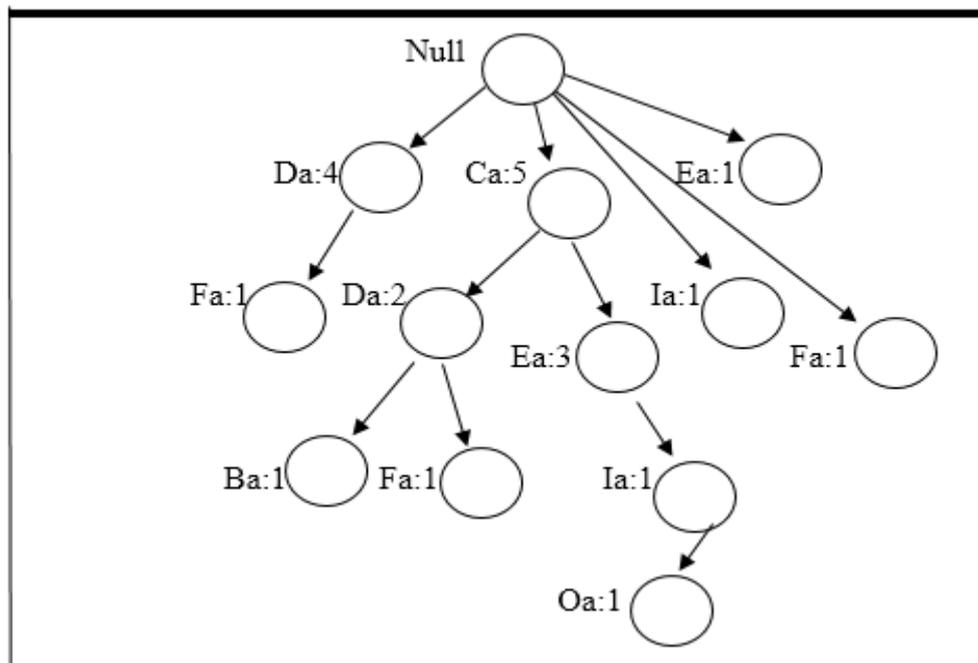
Gambar 3.12 Hasil pembacaan TID 10

Setelah pembacaan TID 10, maka selanjutnya membaca TID 11 yaitu {Ca, Da, Fa} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Da \rightarrow Fa$. Karena memiliki *prefix* yang sama dengan lintasan sebelumnya yaitu item Ca, Da maka lintasan TID 10 bisa ditimpahkan pada lintasan sebelumnya. Setelah itu tambahkan *support count* item Ca yang sebelumnya 4 menjadi 5 dan Da 1 menjadi 2, karena item Ca telah dilewati sebanyak lima kali dan item Da dua kali, sedangkan Fa membuat lintasan sendiri dari lintasan Da dengan nilai support count satu.



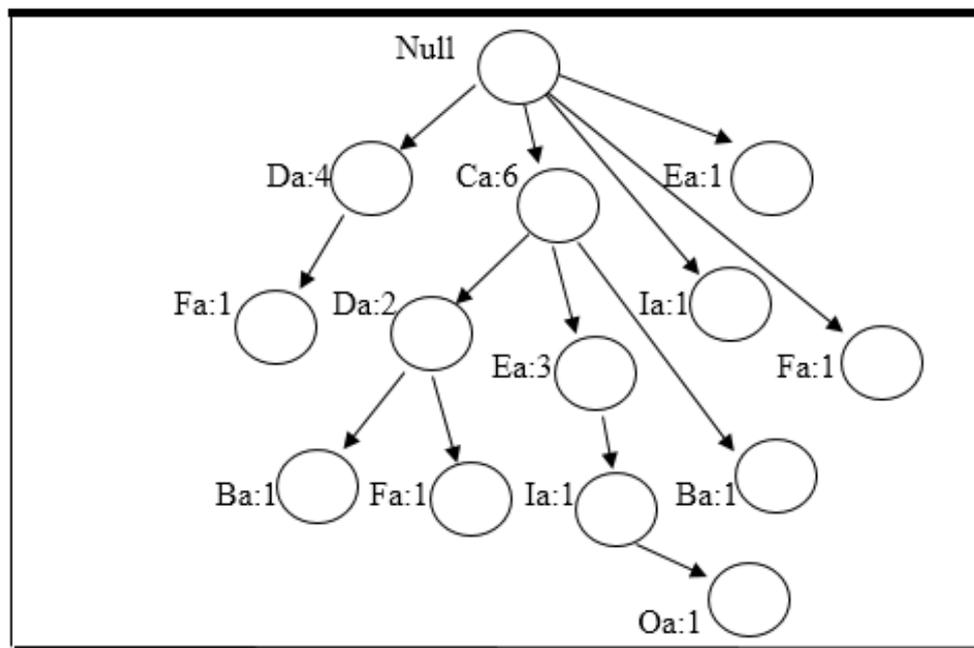
Gambar 3.13 Hasil pembacaan TID 11

Setelah pembacaan TID 11, maka selanjutnya membaca TID 12 yaitu {Da} sehingga membentuk lintasan $\{\} \rightarrow Da$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya, maka TID 12 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count lintasan Da menjadi 4



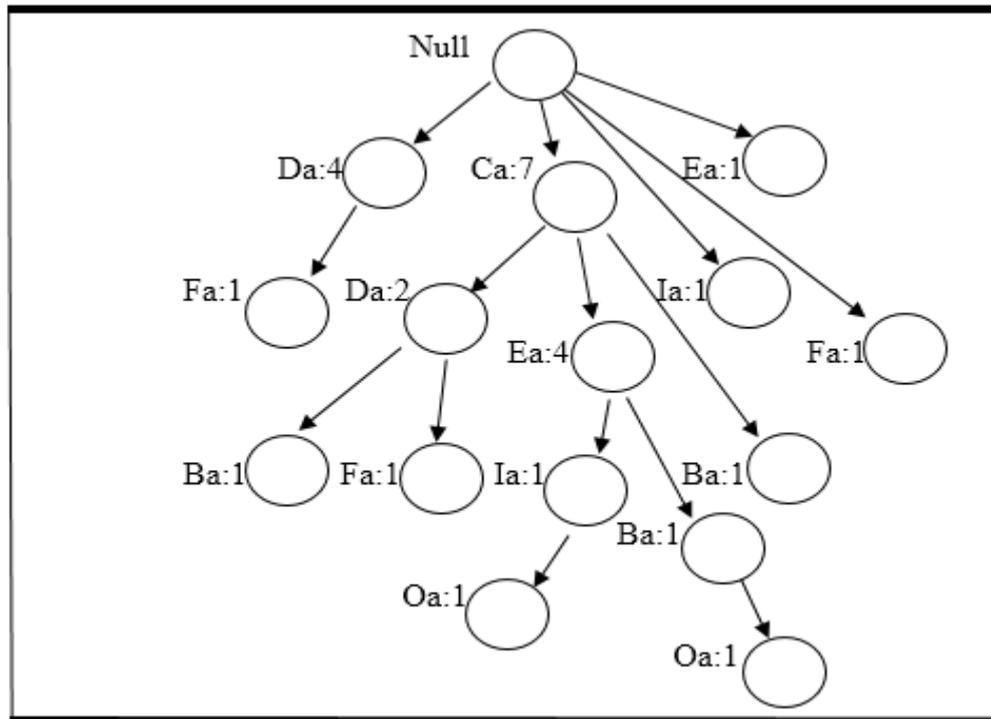
Gambar 3.14 Hasil pembacaan TID 12

Setelah pembacaan TID 12, maka selanjutnya membaca TID 13 yaitu {Ca,Ba} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Ba$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya yaitu Ca, maka TID 12 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count lintasan Ca menjadi 6, sedangkan lintasan Ba membuat lintasan sendiri dari lintasan Ca dengan supcount satu.



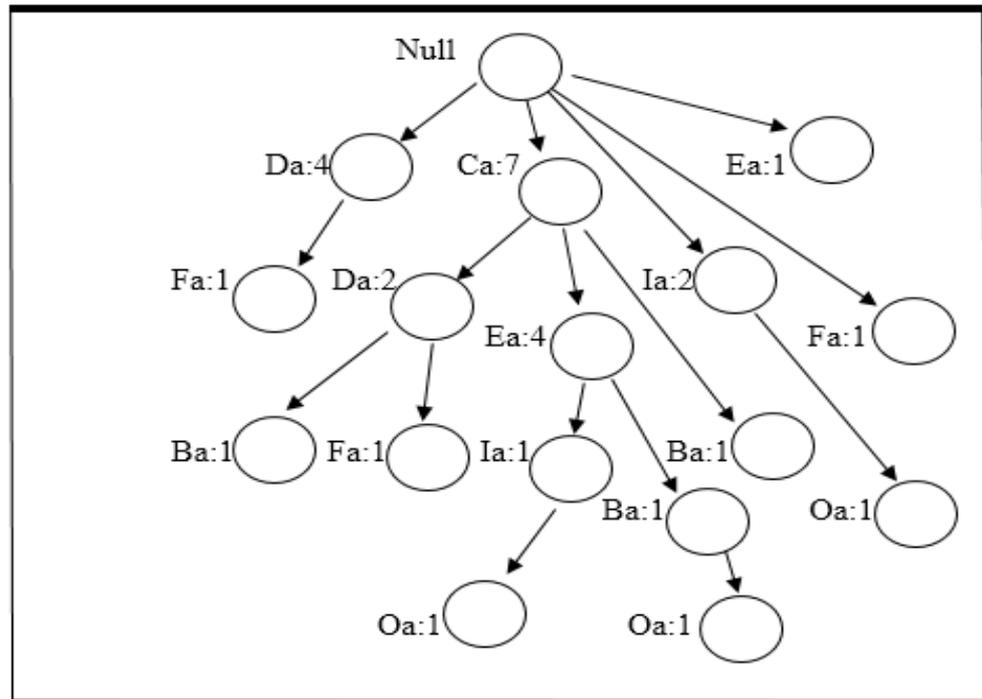
Gambar 3.15 Hasil pembacaan TID 13

Setelah pembacaan TID 13, maka selanjutnya membaca TID 14 yaitu {Ca,Ea,Ba,Oa} sehingga membentuk lintasan $\{\} \rightarrow Ca \rightarrow Ea \rightarrow Ba \rightarrow Oa$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya yaitu Ca,Ea maka TID 14 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count lintasan Ca menjadi 7, dan Ea menjadi 4 sedangkan lintasan Ba→Oa membuat lintasan sendiri dari lintasan Ea dengan supcount satu.



Gambar 3.16 Hasil pembacaan TID 14

Setelah pembacaan TID 14, maka selanjutnya membaca TID 15 yaitu {Ia,Oa} sehingga membentuk lintasan $\{\} \rightarrow Ia \rightarrow Oa$. Karena *prefix* transaksinya ada yang sama dengan lintasan sebelumnya yaitu Ia maka TID 15 dapat di timpahkan kelintasan yang sebelumnya, dan menambah support count lintasan Ia menjadi 2, sedangkan lintasan Oa membuat lintasan sendiri dari lintasan Ia dengan supcount satu.



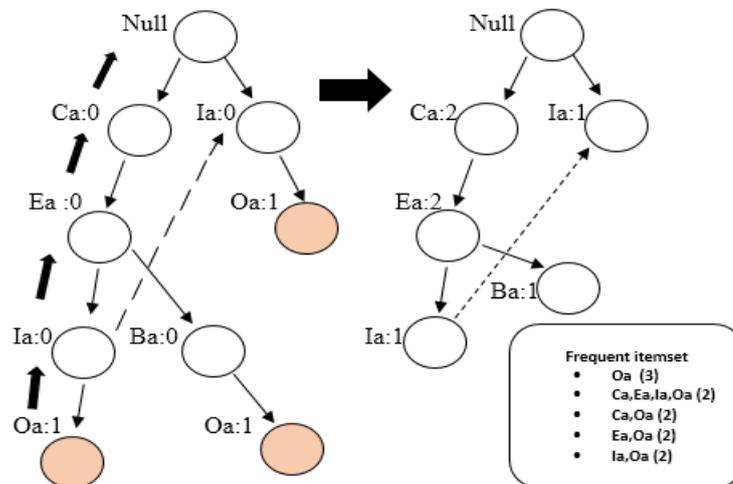
Gambar 3.17 Hasil semua pembacaan ke 15 transaksi

B.2 Tahap penerapan algoritma *FP-Growth*

Setelah pembuatan *FP-Tree* selesai, maka selanjutnya ke tahap algoritma *FP-Growth* untuk mencari semua *subsets* yang memungkinkan dengan cara membangkitkan *conditional FP-Tree* dan mencari *frequent itemset*, dengan menerapkan metode *Divide and Conquer* sesuai urutan *Frequent List*, dilakukan dengan cara mengestrak semua lintasan tiap item dan dilakukan pertama kali dengan item yang paling kecil jumlah kemunculannya yaitu Oa diikuti dengan Ia, Fa, Ba, Ea, Da dan Ca.

1. Kondisi *FP-Tree* untuk item Oa

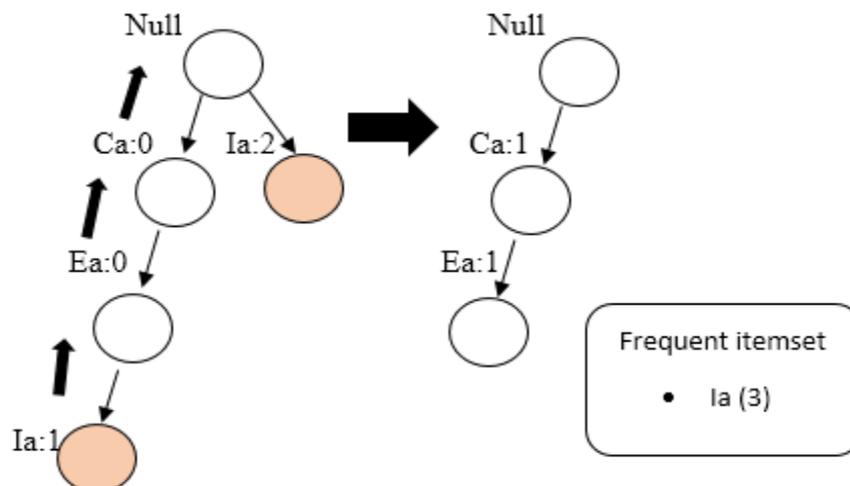
Langkah awal, ekstrak semua lintasan yang hanya berakhir Oa. Selain lintasan Oa diberi nilai 0, hal ini dilakukan agar dapat mengetahui informasi berapa kali *item* yang lain dibeli bersamaan dengan *item* Oa dan bisa mengetahui *frequent itemsets* mana yang memenuhi syarat minimum *support*. Untuk lebih jelasnya, dapat dilihat pada gambar 3.19.



Gambar 3.18 Kondisi *FP-Tree* item Oa

Setelah itu, masukan item Oa sebagai frequent itemset dan naikan satu persatu nilai lintasan Oa ke setiap lintasan yang dilintasi sampai ke null. Pada kasus ini, *item* Ba nilai kemunculan bersamaan dengan *item* Oa hanya 1 kali sehingga *item* Ba dibuang. sedangkan *item* Ia, Ea dan Ca nilai kemunculannya 2 maka *subsets* yang terbentuk adalah {Ca,Ea,Oa}, {Ca,Oa}, {Ea,Oa} {Ia,Oa}.

2. Kondisi *FP-Tree* untuk item Ia

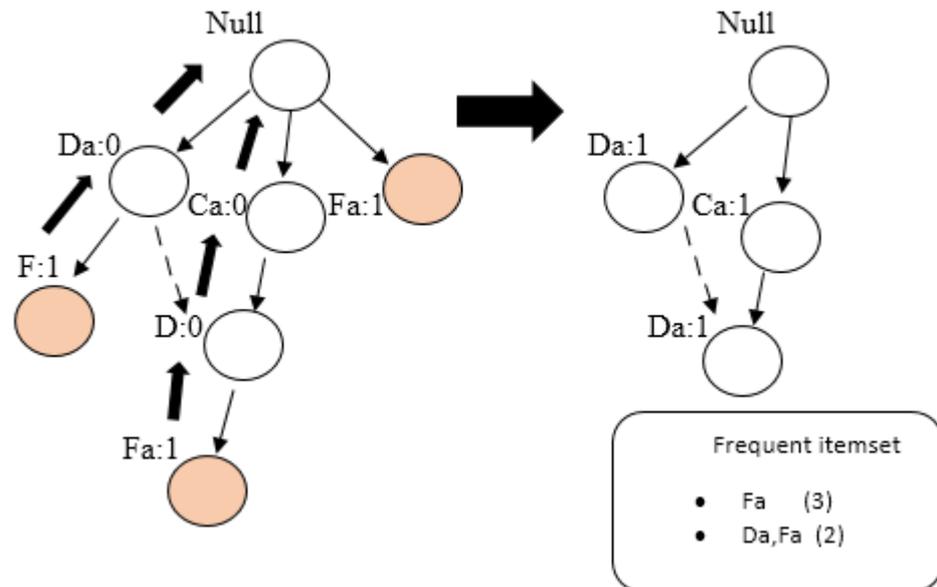


Gambar 3.19 Kondisi *FP-Tree* item Ia

Setelah itu, masukan item I sebagai frequent itemser dan naikan satu persatu nilai lintasan Ia ke setiap lintasan yang dilintasi sampai ke null. Pada

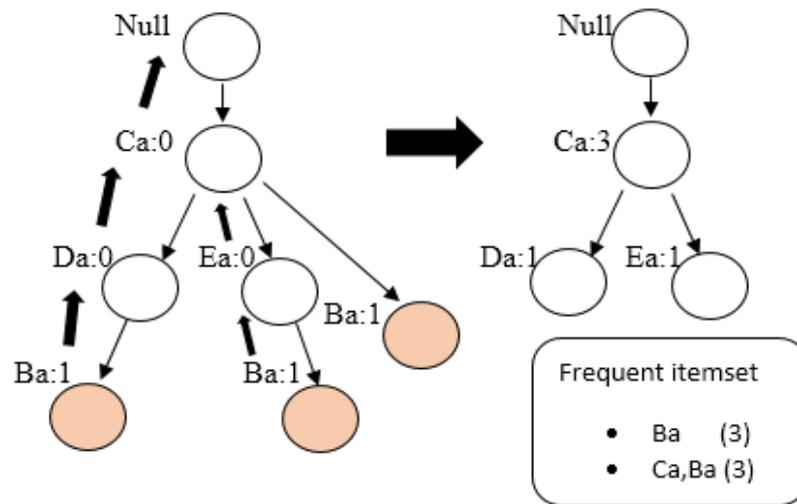
kasus ini, *item* Ea,Ca nilai kemunculan bersamaan dengan *item* Ia hanya 1 kali sehingga *item* Ea,Ca dibuang, Karena *item* Ia berdiri sendiri , maka *subsets* yang terbentuk adalah {Ia}.

3. Kondisi *FP-Tree* untuk *item* Fa

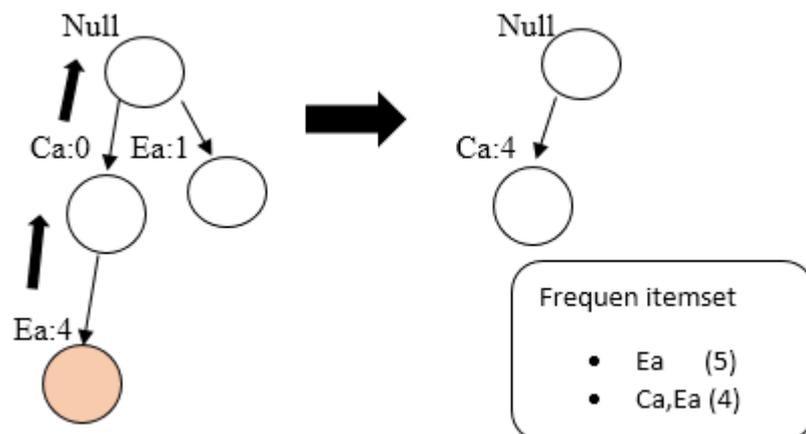


Gambar 3.20 Kondisi *FP-Tree* *item* Fa

Setelah itu, masukan *item* F sebagai frequent itemset dan naikan satu persatu nilai lintasan Fa ke setiap lintasan yang dilintasi sampai ke null. Pada kasus ini, *item* Ca nilai kemunculan bersamaan dengan *item* Fa hanya 1 kali sehingga *item* Ca dibuang, sedangkan *item* Da nilai kemunculan bersamaan dengan *item* Fa 2, maka *subsets* yang terbentuk adalah {Fa},{Da,Fa}.

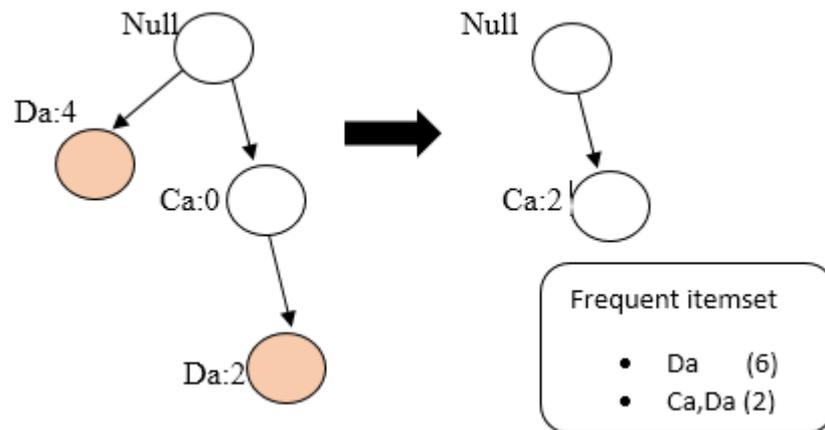
4. Kondisi *FP-Tree* untuk item BaGambar 3.21 Kondisi *FP-Tree* item Ba

Setelah itu, masukan item Ba sebagai frequent itemset dan naikan satu persatu nilai lintasan Ba ke setiap lintasan yang dilintasi sampai ke null. Pada kasus ini, *item* Da,Ea nilai kemunculan bersamaan dengan *item* Ba hanya 1 kali sehingga *item* Da,Ea dibuang, sedangkan item Ca nilai kemunculan bersamaan dengan item Ba 3, maka *subsets* yang terbentuk adalah {Ba},{Ca,Ba}.

5. Kondisi *FP-Tree* untuk item EaGambar 3.22 Kondisi *FP-Tree* item Ea

Setelah itu, masukan item Ba sebagai frequent itemset dan naikan satu persatu nilai lintasan Ea ke setiap lintasan yang dilintasi sampai ke null. Pada kasus ini, *item* Ca nilai kemunculan bersamaan dengan *item* Ea 4 kali, maka *subsets* yang terbentuk adalah {Ea},{Ca,Ea}.

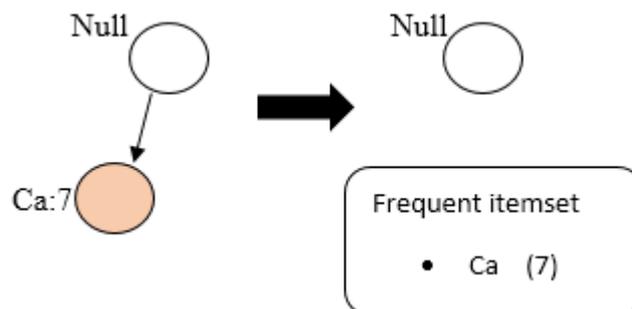
6. Kondisi *FP-Tree* untuk item Da



Gambar 3.23 Kondisi *FP-Tree* item Da

Setelah itu, masukan item Da sebagai frequent itemset dan naikan satu nilai lintasan Da ke setiap lintasan yang dilintasi sampai ke null. Pada kasus ini, *item* Ca nilai kemunculan bersamaan dengan *item* Da 2 kali, maka *subsets* yang terbentuk adalah {Da},{Ca,Da}.

7. Kondisi *FP-Tree* untuk item Ca



Gambar 3.24 Kondisi *FP-Tree* item Ca

Pada kasus ini, lintasan yang berakhir *item* 1 merupakan lintasan tunggal yang berdiri sendiri dan memiliki nilai frekuensi 7 sehingga *frequent itemsets*-nya hanya {1a} yaitu {Ca}.

Setelah menelusuri kondisi *FP-Tree* dari perhitungan algoritma *FP-Growth* didapat 15 *Frequent itemsets* yang hasilnya dirangkum pada tabel 3.8

Tabel 3.8 Hasil *Frequent itemset*

<i>suffix</i>	<i>Frequent itemset</i>
Oa	{{Ca, Oa:2}, {Ea, Oa:2}, {Ca, Ea, Oa:2}, {Ia, Oa:2}, {Oa:3}}
Ia	{Ia:3}
Fa	{{Da, Fa:2}, {Fa:3}}
Ba	{{Ca, Ba:3}, {Ba:3}}
Ea	{{Ca, Ea:4}, {Ea:5}}
Da	{{Ca, Da:2}, {Da:6}}
Ca	{Ca:7}

Dari 15 *Frequent itemsets* tersebut, tidak semua dihitung. Karena dalam menghasilkan *Association Rule*, minimal *Frequent itemsets* yang dihitung terdapat 2 *item* dimana jika membeli *item* Aa maka akan membeli *item* Ba. Sehingga yang layak dihitung *confidence*-nya adalah 8 *subsets*, diantaranya: {Ea,Oa}, {Ca,Oa}, {Ca,Ea,Oa}, {Ia,Oa}, {Da,Fa}, {Ca,Ba}, {Ca,Ea}, {Ca,Da}.

Setelah mendapatkan *frequent itemsets* yang akan dihitung, selanjutnya adalah membuat *rule* dengan menghitung *confidence*-nya. Hanya pola yang nilai *confidence*-nya ≥ 0.75 yang akan diambil, Rumus menghitung *confidence* dapat dilihat pada rumus berikut :

$$confidence(A \cup B) = \frac{support\ count(A \cup B)}{support\ count(A)} \dots \dots \dots$$

Menghitung *confidence* untuk *frequent item set* dengan minimum *confidence* 0,75 didapat pola dan hasil perhitungan sebagai berikut:

1)Frequent itemset {Ea→Oa}

$$Ea \rightarrow Oa = 2/5 = 0,4$$

$$Oa \rightarrow Ea = 2/3 = 0,6$$

2)Frequent itemset {C→O}

$$Ca \rightarrow Oa = 2/7 = 0,28$$

$$Oa \rightarrow Ca = 2/3 = 0,6$$

3)Frequent itemset {Ca→Ea→Oa}

$$Oa \wedge Ea \rightarrow Ca = 2/2 = 1$$

$$Ca \rightarrow Oa \wedge Ea = 2/7 = 0,28$$

$$Ea \rightarrow Oa \wedge Ca = 2/5 = 0,4$$

$$Oa \wedge Ca \rightarrow Ea = 2/2 = 1$$

$$Oa \rightarrow Ea \wedge Ca = 2/3 = 0,6$$

$$C, E \rightarrow O = 2/4 = 0,5$$

- 4)Frequent itemset {Ia→Oa}
 $Oa \rightarrow Ia = 2/3 = 0,6$
 $Ia \rightarrow Oa = 2/3 = 0,6$
- 5)Frequent itemset {Da→Fa}
 $Da \rightarrow Fa = 2/6 = 0,3$
 $Fa \rightarrow Da = 2/3 = 0,6$
- 6)Frequent itemset {Ca→Ba}
 $Ca \rightarrow Ba = 3/7 = 0,4$
 $Ba \rightarrow Ca = 3/3 = 1$
- 7)Frequent itemset {Ca→Ea}
 $Ca \rightarrow Ea = 4/7 = 0,5$
 $Ea \rightarrow Ca = 4/5 = 0,8$
- 8)Frequent itemset {Ca→Da}
 $Ca \rightarrow Da = 2/7 = 0,28$
 $Da \rightarrow Ca = 2/6 = 0,3$

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 inisial produk akan dikembalikan sesuai nama dari masing-masing produk agar pembacaan *rule* lebih jelas. Adapun *rule* yang di dapat dirangkum pada tabel 3.8

Tabel 3.9 Rule yang memenuhi

<i>Rule</i>	<i>confidence</i>
{Waterfall, Aleksander} → {Esther}	1,0%
{Waterfall, Esther} → {Aleksander}	1,0%
{Antha cuba} → {Esther}	1,0%
{Aleksander} → {Esther}	0,8%

Dari perhitungan *confidence* diatas terdapat 4 pola kebiasaan konsumen dalam berbelanja produk yang memenuhi syarat *confidence* $\geq 0,75$ yaitu {Oa^Ea→Ca}, {Oa^Ca→Ea}, {Ba→Ca}, dan {Ea→Ca}.

1. {Oa^Ea→Ca}, yang artinya (jika konsumen membeli **Waterfall** dan **Aleksander** maka membeli **Esther** dengan nilai *confidence* 1,0).

2. $\{Oa \wedge Ca \rightarrow Ea\}$, yang artinya (jika konsumen membeli **Waterfall** dan **Esther** maka membeli **Alexsander** dengan nilai *confidence* 1,0)
3. $\{Ba \rightarrow Ca\}$, yang artinya (jika konsumen membeli **Antha cuba** maka membeli **Esther** dengan nilai *confidence* 1.0)
4. $\{Ea \rightarrow Ca\}$, Yang artinya (jika konsumen membeli **Alexsander** maka membeli **Esther** dengan nilai *confidence* 0,8)

Jadi dari ke 15 transaksi bulan januari 2014, dengan membatasi nilai minimum *support* 0,3 dan minimum *confidence* 0,75 didapat 4 pola kebiasaan konsumen dalam berbelanja dengan artian didapat 4 produk yang paling sering di beli oleh konsumen, yaitu produk yang berinisial **Ca Esther, Oa Waterfall, Ea Alexsander** dan **Ba Antha Cuba**, sehingga dapat ditentukan ketiga produk inilah yang dapat dikatakan sebagai produk terunggul dari 1 bulan januari dengan 15 transaksi.

3.4 Perancangan sistem

Pada bagian ini akan menjelaskan mengenai Diagram Konteks (*Context Diagram*), Diagram berjenjang, Diagram Alir Data (*Data Flow Diagram*), Desain Basis Data (*Database*), Desain Antar Muka (*Interface*), berikut ini penjelasan dari sub bab tersebut.

3.4.1 Diagram Konteks (*Context Diagram*)

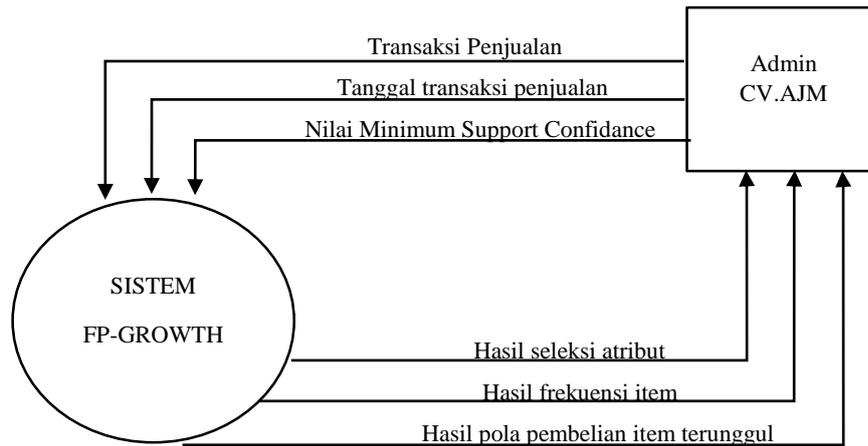
Berdasarkan dari diagram alir kerja maka dapat dimodelkan sebuah diagram konteks (*Context Diagram*) sistem berbasis web untuk menentukan pola kebiasaan pelanggan dalam berbelanja yang dalam hal ini berfungsi sebagai gambaran hubungan antara entitas luar, masukan dan keluaran sistem.

Dalam diagram konteks **Gambar 3.3** merupakan gambaran sistem secara garis besar, dimana terdapat dua entitas luar yang berhubungan dengan sistem, yaitu :

1. Admin CV.AJM merupakan entitas yang memasukan data transaksi penjualan dan memilih range tanggal data transaksi yang akan di proses

serta menentukan nilai minimum support dan minimum confidence yang digunakan sebagai acuan dalam menentukan pola kebiasaan pelanggan dalam berbelanja.

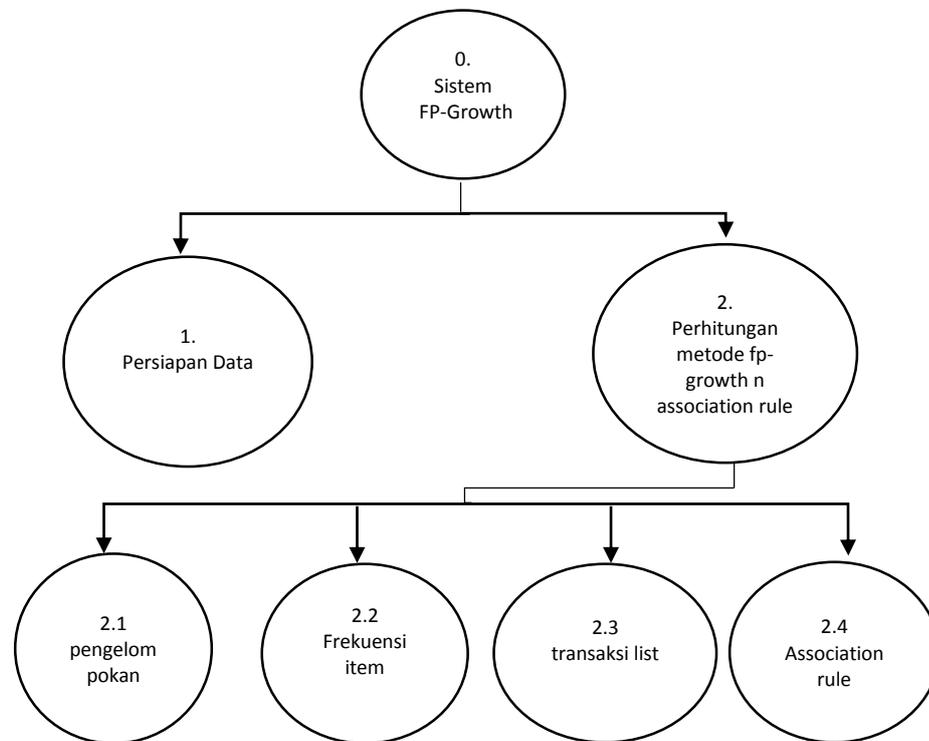
Berikut **Gambar 3.3** adalah Diagram Konteks sistem berbasis web untuk menentukan pola kebiasaan pelanggan dalam berbelanja. :



Gambar 3.25 Diagram Konteks Sistem *FP-Growth*

3.4.2 Diagram Berjenjang

Diagram berjenjang proses merupakan diagram yang menjelaskan secara keseluruhan blok proses yang ada pada sistem. Terdapat 3 proses yang dilakukan user yaitu proses persiapan data, proses penerapan metode fp-growth, dan proses laporan hasil pola yang didapat, untuk proses penerapan metode fp-growth sendiri terdapat 4 proses diantaranya pengelompokan data, menghitung jumlah frekuensi, pengurutan berdasarkan frekuensi item, dan menentukan pola yang dihasilkan. Diagram berjenjang proses dapat dilihat pada gambar 3.26



Gambar 3.26 Diagram Berjenjang menentukan frekuensi

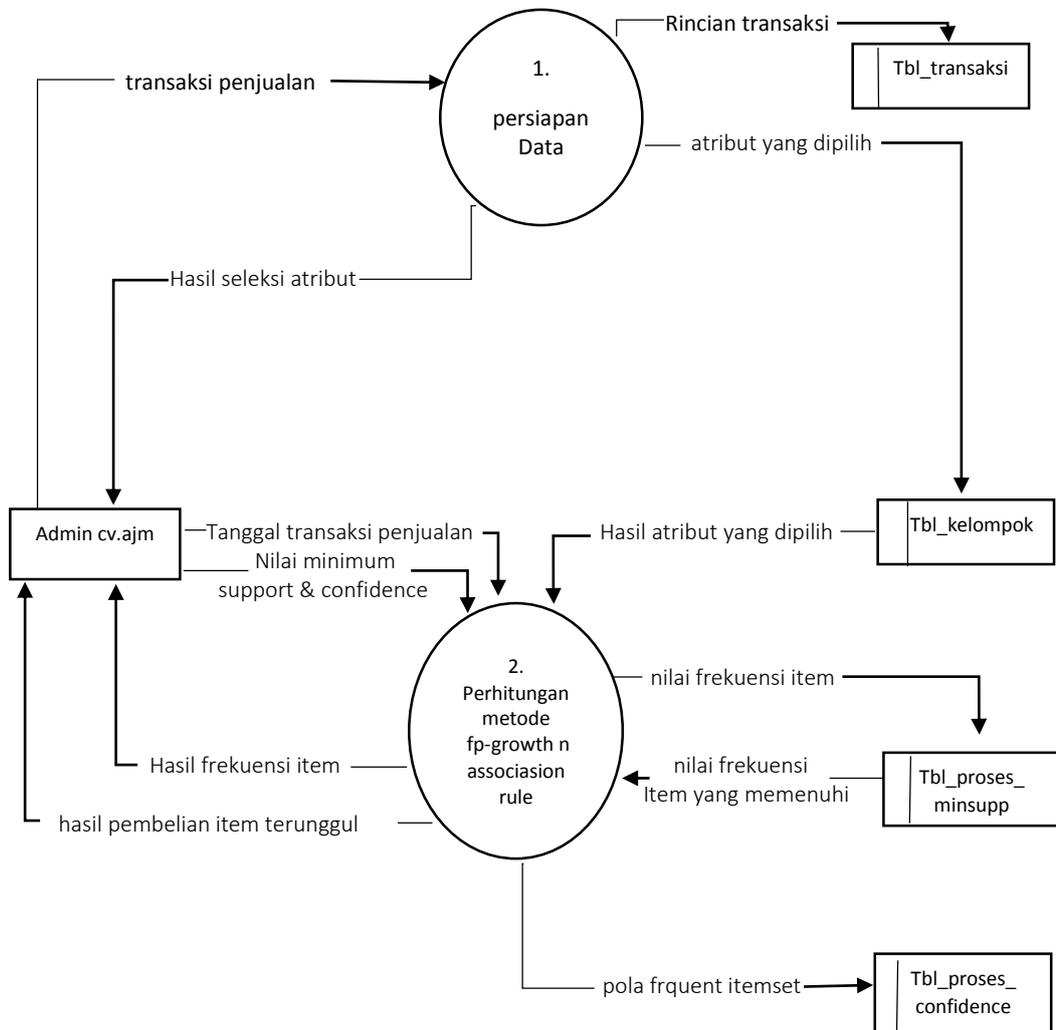
3.4.3 Diagram Alir Data (*Data Flow Diagram*)

DFD (*Data Flow Diagram*) merupakan representasi grafik dari keseluruhan proses yang menggambarkan aliran informasi yang di aplikasikan sebagai data yang mengalir dari *input* dan *output* oleh sistem.

3.4.3.1 DFD Level 0

Berikut **Gambar 3.27** merupakan DFD Level 0 yang menjelaskan seluruh proses yang terjadi dalam sistem untuk menentukan produk terunggul CV. Anugrah Jaya Muliya :

3.4.3.1 DFD level 0



Gambar 3.27 DFD Level 0

Adapun rincian DFD Level 1 seperti berikut

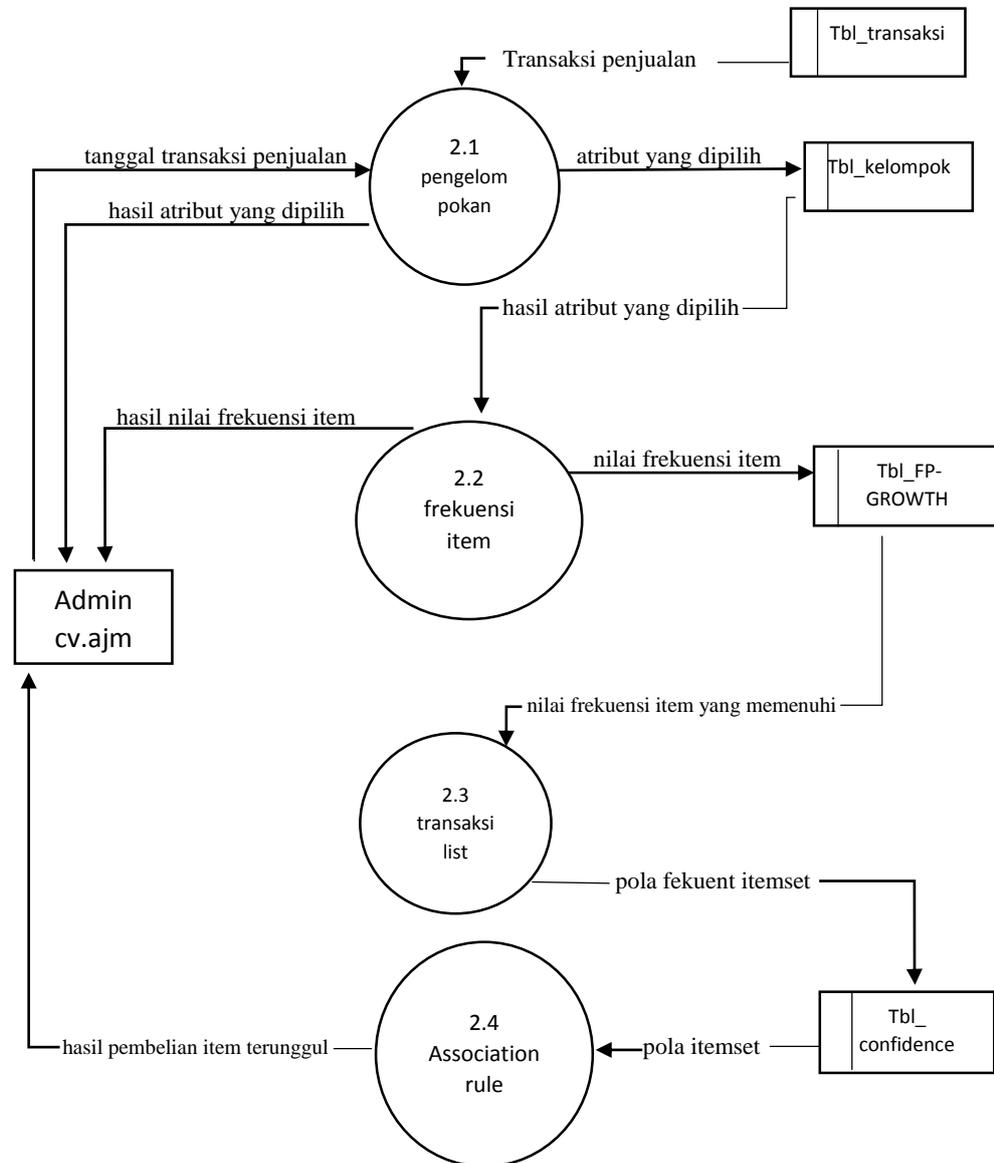
1. Proses 1 persiapan data, user memasukan data transaksi penjualan yang akan tersimpan pada tabel transaksi, dimana didalam proses persiapan data ini meliputi proses seleksi data yang memilih atribut dari data transaksi, yang nantinya akan di simpan ke tabel kelompok, dan menghasilkan hasil seleksi atribut.

2. Proses 2 perhitungan metode *FP-Growth* dan *association rule*, dari data yang sudah siap untuk diproses, Pertama user akan menentukan tanggal transaksi, untuk diketahuinya jumlah frekuensi kemunculan item, yang akan tersimpan pada tabel proses minimum support, setelah didapat jumlah kemunculan item, user memasukan batasan nilai minimum *support* dan *confidence* untuk menentukan frekuensi item set yang memenuhi minimum nilai batasan yang ditentukan, dan akan tersimpan pada tabel proses confidence, Kemudian frekuensi yang memenuhi minimum support akan dihitung confidencenya oleh sistem untuk mendapatkan hasil rule atau pola yang diharapkan, dan menghasikan informasi hasil ahir pola yang didapat.

3.4.3.2 DFD Level 1

DFD level 1 untuk proses perhitungan dan penentuan produk terunggul dengan metode *FP-Growth* dimana didalamnya terdapat 4 proses yaitu pengelompokan data, frekuensi item, mengurutkan item sesuai frekuensi yang disebut transaksi list, dan *Association rule*, melakukan perhitungan confidence untuk menentukan pola item yang terbaik. Untuk lebih jelas proses dapat dilihat pada **Gambar 3.28**

3.5.3.2 DFD Level 1



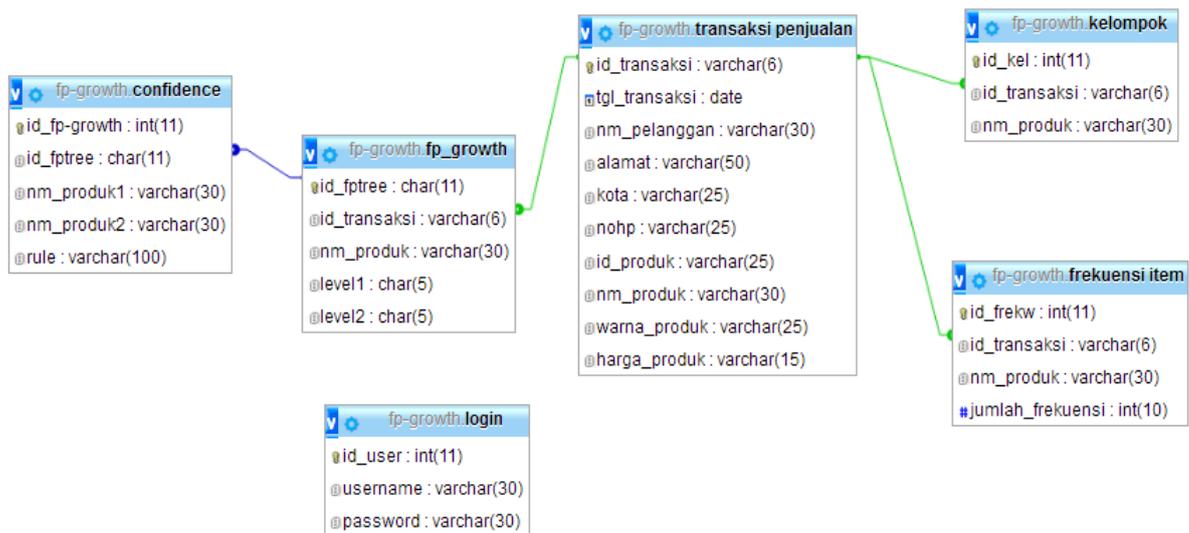
Gambar 3.28 DFD Level 1

3.6 Perancangan Basis Data

Perancangan *database* menjadi hal yang sangat utama dalam pembuatan sebuah sistem *FP-Growth* untuk menentukan pola kebiasaan konsumen dalam berbelanja, dimana nanti akan terdapat struktur tabel utama dan pendukung serta relasi tabel yang akan menghubungkan tabel yang satu dengan tabel lainnya.

3.6.1 Diagram Relasi

Diagram relasi berfungsi untuk menggambarkan relasi antar tabel yang mempunyai atribut kunci utama yang sama, sehingga tabel-tabel tersebut menjadi suatu kesatuan yang dihubungkan oleh kunci tersebut. Berikut adalah skema relasi tabel dalam sistem *FP-Growth* untuk menentukan pola konsumen dalam berbelanja dapat dilihat pada **Gambar 3.29** :



Gambar 3.29 Diagram relasi penentuan pola frekuensi itemset

3.6.2 Desain struktur tabel

Desain tabel database pada sistem *FP-Growth* untuk menentukan produk garmen terunggul CV.Anugrah Jaya Mulia sebagai berikut :

1. Tabel login user

Tabel user ini dibuat untuk secara khusus agar bisa mengakses aplikasi ini. Data dari user tersebut tersimpan dalam tabel user. Struktur dari tabel user dapat dilihat pada tabel 3.10

Tabel 3.10 login user

Nama field	Type Data	Leght	Ket
Id_user	Varchar	30	Primary key
Username	Varchar	30	
Password	Varchar	30	

2. Tabel data transaksi

Tabel data transaksi, yang berfungsi untuk menyimpan data transaksi penjualan, yang akan diolah untuk dicari pola item atau produk yang sering dibeli oleh konsumen, struktur tabel data transaksi dapat dilihat pada tabel 3.11

Tabel 3.11 data transaksi

Nama field	Type data	Leght	key
Tgl_transaksi	varchar	6	
Id_transaksi	int	6	Primary key
Nm_pelanggan	Varchar	30	
Alamat	Varchar	50	
Kota	Varchar	25	
Nohp	Varchar	25	
Id_produk	Varchar	25	
Nama_produk	Varchar	25	
Harga_produk	Varchar	15	

3. Tabel data kelompok

Tabel data kelompok, berfungsi untuk penyimpanan data atribut yang sudah diseleksi dan dikelompokan, struktur tabel klompok dapat dilihat pada tabel 3.12

Tabel 3.12 data kelompok

Nama field	Type data	Leght	key
Id_kel	Varchar	30	Primary key
Id_transaksi	Int	6	
nama_produk	Varchar	25	

4. Tabel data *FP-Growth*

Tabel *FP-Growth*, berfungsi sebagai penyimpanan data item yang memenuhi batasan nilai minimum support untuk di proses perhitungan association rule, struktur tabel dapat dilihat pada tabel 3.13

Tabel 3.13 Data *FP-Growth*

Nama field	Type data	Leght	key
Id_fptree	Varchar	30	Primary key
Id_transaksi	Varchar	30	
Nama_produk	Varhhar	30	
Level 1	char	5	
Levell 2	char	5	

5. Tabel Confidence

Tabel pola item, berfungsi untuk menyimpan data hasil pola perhitungan association rule yang memenuhi batasan nilai minimum *confidence*, struktur tabel pola item dapat dilihat pada tabel 3.14

Tabel 3.14 Confidence

Nama field	Type data	Leght	key
Id_fpgrowth	Varchar	30	Primary key
Nama_produk 1	Varchar	30	
Nama_produk2	Varchar	30	
Rule	Varchar	100	

3.7 Desain antarmuka

Merupakan suatu perancangan antarmuka aplikasi yang digunakan untuk berinteraksi langsung dengan pengguna sistem.

3.7.1 Form Login

Form login adalah halaman awal sebelum *user* dapat menggunakan aplikasi. Pada halaman ini *user* harus mengisi form username, password dan memilih level yang sudah disediakan. Desain *form login* dapat dilihat pada gambar 3.30

Gambar 3.30 Desain Form Login

3.7.2 Form data menu

Pada bagian form data menu adalah bagian inti untuk memproses algoritma *frequent pattern growth*. Pada bagian ini terdapat beberapa proses untuk mengolah data *fp growth* untuk menentukan item terunggul, desain form data menu dapat dilihat pada gambar 3.31

Gambar 3.31 Desain Form Data Menu

3.7.3 Form Menu Petunjuk

Rancangan form menu petunjuk untuk menampilkan petunjuk dan aturan cara menggunakan sistem FP-Growth, desain form petunjuk dapat dilihat pada gambar 3.32

Gambar 3.32 Desain Form menu petunjuk

3.7.4 Form Menu Input Data Transaksi Penjualan

Rancangan dari form Input Data Transaksi Penjualan digunakan untuk memasukkan data penjualan yang akan diolah dan di proses ke perhitungan *FP-Growth*, desain dapat dilihat pada gambar 3.33

Nama barang	qty	harga jual rp	sub total rp

Gambar 3.33 Desain Menu Form Input Data Transaksi Penjualan

3.7.5 Form Menu Data Transaksi

Rancangan form menu data transaksi menampilkan data transaksi yang tersimpan, desain menu data transaksi dapat dilihat pada gambar 3.34

The screenshot shows a web application interface for 'SISTEM FP-GROWTH CV.AJM'. The main content area is titled 'DATA TRANSAKSI PENJUALAN' and contains a table with the following data:

No.faktur	Tgl.jual	Nm.customer	
Pj-001/2014	01/01/2014	Febrinia rachmasiam	hapus

Each row in the table has a 'view' button under the 'No.faktur' column and a 'hapus' button under the empty column. A 'MENU APLIKASI' dropdown menu is open in the top right corner, listing the following options: 'Menu utama', 'Input data barang', 'Input data customer', and 'Keluar'. The interface also features a horizontal scrollbar at the bottom of the table area.

Gambar 3.34 Desain Form Menu Data Transaksi

3.7.6 Form Menu Perhitungan *FP-Growth*

Rancangan dari form proses *FP-Growth* untuk menentukan frequent itemset dibagi menjadi 4 bagian tab, dan terdapat form untuk menentukan tanggal dan memasukan batasan minimum support dan minimum confidence, seperti yang terlihat pada *interface* berikut ini:

1. Tab proses pengelompokan

Tab proses pengelompokan, merupakan rancangan halaman interface untuk menampilkan hasil data yang sudah ditransformasi dan dikelompokan, desain tab proses pengelompokan dapat dilihat pada gambar 3.35

SISTEM FP-GROWTH CV.AJM

PERHITUNGAN FP-GROWTH

Tanggal : s/d Minimum support : %
 Minimum confidence : %

DATA KELOMPOK

TID	item
1	Alexander, esther
2	Anta cuba, esther, adela top
3	Donna, arlinaa
4	Camara, febby sries 1
5	Berete, marta

Gambar 3.35 Desain proses pengelompokan data

2. Tab proses frekuensi item

Tab proses frekuensi item, merupakan rancangan halaman untuk menampilkan jumlah kemunculan item atau frekuensi item, desain tab proses frekuensi item dapat dilihat pada gambar 3.36

SISTEM FP-GROWTH CV.AJM

PERHITUNGAN FP-GROWTH

Tanggal : s/d Minimum support : %
 Minimum confidence : %

DATA FREKUENSI ITEM

item	frekuensi
alexander	1
esther	2
Anta cuba	1
adela	1
marta	1
berete	2
donna	2

Gambar 3.36 Desain proses frekuensi item

3. Tab proses data transaksi list

Tab proses data transaksi list, merupakan rancangan halaman untuk menampilkan data list, data item yang diubah sesuai frekuensi item yang memenuhi minimum *support*, desain dapat dilihat pada gambar 3.37

SISTEM FP-GROWTH CV.AJM

PERHITUNGAN FP-GROWTH

Tanggal : s/d Minimum support : %
 Minimum confidence : % PROSES

DATA KELOPOK LIST / DATA YANG MEMENUHI MIN SUPP

tid	item
1	
2	
3	
4	
5	

Gambar 3.37 Desain proses data transaksi list

4. Tab proses hasil pola

Tab proses hasil pola, merupakan rancangan halaman untuk menampilkan hasil pola item yang didapat, desain dapat dilihat pada gambar 3.38

SISTEM FP-GROWTH CV.AJM

PERHITUNGAN FP-GROWTH

Tanggal : s/d Minimum support : %
 Minimum confidence : % PROSES

DATA POLA ITEM

Pola item	confidence
Axesader→ester	1%
Ester→anta cuba	1%
Axesander^antacuba→estre	0,8%

Gambar 3.38 Desain proses hasil pola

3.8 Skenario dan Evaluasi

3.8.1 Skenario Pengujian Sistem

Skenario pengujian sistem dilakukan dengan memasukkan minimum *support confidence* untuk membatasi data transaksi yang akan diuji, data tersebut diperoleh dari CV. Anugrah Jaya Mulia Gunung Anyar Surabaya.

Dalam melakukan pengujian dari data transaksi digunakan 2 macam atribut meliputi, id transaksi dan nama produk yang dibeli oleh konsumen. Pengujian dilakukan dengan mengambil contoh (sampel) data untuk membentuk suatu *FP-Tree*. Sampel data yang diambil sebanyak jumlah transaksi yang ditentukan, dengan minimum *support* sebesar 0.3 atau sama dengan 30%. Karena data transaksi yang digunakan dalam sampel sebanyak 15 data transaksi, penentuan minimum *confidence* sebesar 0.75 atau sama dengan 75%. Selanjutnya hasil dari setiap pengujian akan dilakukan evaluasi sistem dengan metode *Divides and conquer* dan dihitung nilai *confidence* nya.

Hasil pengujian tersebut kemudian dibandingkan dengan hasil perhitungan secara manual sebelumnya untuk mengetahui tingkat keakuratan sistem.

3.9 Spesifikasi dan Kebutuhan Sistem

Dalam pembuatan sistem penentuan frequent itemset CV. Anugrah Jaya Mulia dengan metode FP-Growth dibutuhkan spesifikasi perangkat keras dan perangkat lunak.

1. Kebutuhan Perangkat Keras

Perangkat keras adalah komponen fisik peralatan yang membentuk sistem komputer, serta peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun minimal perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini adalah sebagai berikut :

1. Prosesor Intel Pentium Dual-Core 2.2 Ghz
2. Memory RAM 2 GB

3. Monitor VGA atau SVGA 14 inch
4. Harddisk 320 GB
5. Keyboard
6. Mouse

2. Kebutuhan Perangkat Lunak

Sedangkan untuk *spesifikasi software* (kebutuhan perangkat lunak) untuk merancang aplikasi ini adalah sebagai berikut :

1. Sistem Operasi Windows 7
2. Microsoft Office 2007
3. Microsoft Office Visio 2007
4. XAMPP Control Panel V3.21
5. Sublime versi 3.0
6. Mozilla Firefox
7. SQLyog Ultimate