ALGORITMA A PRIORI

- Jenis aturan asosiasi:
  - A priori
  - Generalized Rule Induction
  - Algoritma Hash Based
- Association rule mining merupakan teknik data mining utk menemukan aturan asosiatif antara suatu kombinasi item.

Contohnya:

- Analisis pembelian di suatu pasar swalayan, dapat diketahui brp besar kemungkinan seorang pelanggan membeli roti bersamaan dgn susu.
- Dgn kondisi tsb:
  - Pemilik swalayan dpt mengatur penempatan barangnya, atau
  - Merancang promosi pemasaran dgn memakai kupon diskon utk kombinasi barang tertentu.

- Association Rule menjadi terkenal karena aplikasinya utk menganalisis isi keranjang belanja pelanggan di pasar swalayan
- Makanya itu sering disebut Market Basket Analysis
- AR jg dikenal salah satu teknik DM yg menjadi dasar dari berbagai teknik DM lainnya.
- Salah satu tahap analisis asosiasi yg menarik perhatian byk peneliti utk menghasilkan algoritma yg efisien adalah analisis pola frekuensi tinggi (*frequent pattern mining*)

- Penting tidaknya suatu aturan asosiatif dapat diketahui dgn dua parameter
  - Support
  - Confidence

- Support (nilai penunjang) → persentase kombinasi item tsb dlm database.
- Confidence (nilai kepastian) → kuatnya hubungan antar item dalam aturan asosiasi

CONTOH

{roti, mentega}→ {susu} (support=40%, confidence = 50%)

Aturan tsb artinya ...

- 50% dari transaksi database yg memuat item roti dan mentega juga membeli susu. 40% dari seluruh transaksi yg ada di database memuat ketiga item tsb
- "Seorang pelanggan yg membeli roti dan mentega punya kemungkinan 50% utk juga membeli susu. Aturan tsb cukup signifikan karena mewakili 40% dari catatan transaksi selama ini"

METODOLOGI DASAR ANALISIS ASOSIASI

1. Analisis Pola Frekuensi Tinggi

Tahap ini mencari kombinasi item yg memenuhi syarat minimmum dari nilai support dalam database. Nilai support sebuah item diperoleh dengan rumus sbb:

$$\text{Support (A)} = \frac{\text{Jumlah transaksi mengandung A}}{\text{Transaksi Total}}$$

- Nilai support dari 2 item diperoleh dari rumus 2 berikut:

$$\text{Support } (A,B) = P(A \cap B)$$

$$\text{Support } (A,B) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi}}$$

- Sebagai contoh, terdpt database dari transaksi belanja pasar swalayan sbb:

Transaksi

Transaksi	Item yg dibeli
1	Susu, Teh, Gula
2	Teh, Gula, Roti
3	Teh, Gula
4	Susu, Roti
5	Susu, Gula, Roti
6	Teh, Gula
7	Gula, Kopi, Susu
8	Gula, Kopi, Susu
9	Susu, Roti, Kopi
10	Gula, Teh, Kopi

REPRESENTASI DALAM BENTUK DATABASE TRANSAKSIAONAL

Transaksi	Item yg dibeli
1	Susu
1	Teh
1	Gula
2	Teh
2	Gula
2	Roti
3	Teh
3	Gula
4	Susu
4	Roti
5	Susu
5	Gula
5	Roti

Transaksi	Item yg dibeli
6	Teh
6	Gula
7	Gula
7	Kopi
7	Susu
8	Gula
8	Kopi
8	Susu
9	Susu
9	Roti
9	Kopi
10	Gula
10	Teh
10	Kopi

Format Tabular

Transaksi	Teh	Gula	Kopi	Susu	Roti
1	1	1	0	1	0
2	1	1	0	0	1
3	1	1	0	0	0
4	0	0	0	1	1
5	0	1	0	1	1
6	1	1	0	0	0
7	0	1	1	1	0
8	0	1	1	1	0
9	0	0	1	1	1
10	1	1	1	0	0

Misalkan,

- D adalah himpunan transaksi yg dipresentasikan dlm tabel "Transaksi" diatas
- Dimana setiap transaksi T dalam D mempresentasikan himpunan item yg berada dalam I
- I adalah himpunan item yg dijual {teh, gula, kopi, susu, roti}
- Jika kita memiliki himpunan item A (misal Susu dan Gula) dan himpunan lain B (Misal Kopi)

  Maka atusan asosiasi akan berbentuk:

  Jika A, maka B (A → B)

- Dimana *antecedent* A dan *consequent* B merupakan *subset* dari I, dan A dan B merupakan *mutually exclusive* di mana aturan:

  Jika A, maka B tidak berarti Jika B, maka A

- Definisi diatas tdk berlaku utk aturan trivial spt: *Jika Beans dan Squash, maka Beans*
- Seorang analis mungkin saja hanya akan mengambil aturan yg memiliki support dan/atau confidence yg tinggi.
- Aturan yg kuat adalah aturan2 yg melebihi kriteria support dan/atau confidence minimum.
- Misalnya, seorang analis menginginkan aturan yg memiliki support lebih dari 20% dan confidence lebih dari 35%.

- Sebuah itemset adalah himpunan item-item yg ada dlm I
- k-itemset adalah itemset yg berisi k item, misalnya:
  - {Teh, Gula} adalah sebuah 2-itemset
  - {Teh, Gula, Roti} merupakan 3-itemset
- Frequent Itemset menunjukkan itemset yg memiliki frekuensi kemunculan l
- Lebih dari nilai minimum yg telah ditentukan (Φ), misalkan:
  - Φ =2, maka semua itemset yg frekuensi kemunculannya lebih dari atau sama dgn 2 kali disebut frequent.
  - Himpunan dari frequent k-itemset dilambangkan dengan F_k

KOMBINASI	JUMLAH
Teh, Gula	5
Teh, Kopi	1
Teh, Susu	1
Teh, Roti	1
Gula, Kopi	3
Gula, Susu	4
Gula, Roti	2
Kopi, Susu	3
Kopi, Roti	1
Susu, Roti	3

CALON 2-ITEMSET

Dari data tsb, jika ditetapkan nilai Φ =2, maka:

F_2 = {{Teh, Gula}, {Gula, Kopi}, {Gula, Susu}, {Gula, Roti}, {Kopi, Susu}, {Susu, Roti}}

CALON 3-ITEMSET

KOMBINASI	JUMLAH
Teh, Gula, Kopi	1
Teh, Gula, Susu	1
Gula, Susu, Kopi	2
Gula, Susu, Roti	0
Gula, Kopi, Roti	0
Kopi, Susu, Roti	1

Dari calon 3-itemset dari tabel tsb.,
Terpilih F3 = {{Gula, Susu, Kopi}}
karena hanya kombinasi tsb
yg memiliki
Frekuensi kemunculan >= Φ

2. Pembentukkan Aturan Asosiasi

- Setelah semua pola frekuensi tinggi ditemukan, kemudian dicari aturan asosiasi yg memenuhi syarat minimum untuk confidence dengan menghitung confidence aturan asosiatif A → B
- Nilai confidence dari aturan A → B diperoleh dari rumus berikut:

$$\text{Confidence} = P(B \mid A) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi mengandung A}}$$

- Dari F3 yg telah ditemukan, dpt dilihat besarnya nilai support dan confidence dari calon aturan asosiasi.

Aturan	Confidence	
Jika membeli Gula dan Susu, maka akan membeli Kopi	2/4	50 %
Jika membeli Gula dan Kopi, maka akan membeli Susu	2/3	67 %
Jika membeli Kopi dan Susu, maka akan membeli Gula	2/3	67 %

- Misalnya, ditetapkan nilai confidence minimal 60%, maka aturan yg bisa terbentuk adalah dengan 2 antecedent berikut:
  - Jika membeli Gula dan Kopi, maka akan membeli Susu
  - Jika membeli Kopi dan Susu, maka akan membeli Gula

Transaksi

Transaksi	Item yg dibeli
1	Susu, Teh, Gula
2	Teh, Gula, Roti
3	Teh, Gula
4	Susu, Roti
5	Susu, Gula, Roti
6	Teh, Gula
7	Gula, Kopi, Susu
8	Gula, Kopi, Susu
9	Susu, Roti, Kopi
10	Gula, Teh, Kopi

CALON ATURAN ASOSIASI DARI F2

Aturan	Confidence	
Jika membeli teh, maka membeli gula	5/5	100 %
Jika membeli gula, maka membeli teh	5/8	62.5 %
Jika membeli gula, maka membeli kopi	3/8	37.5 %
Jika membeli kopi, maka membeli gula	3/4	75 %
Jika membeli gula, maka membeli susu	4/8	50 %
Jika membeli susu, maka membeli gula	4/6	67 %
Jika membeli gula, maka membeli roti	2/8	25 %
Jika membeli roti, maka membeli gula	2/4	50 %
Jika membeli kopi, maka membeli susu	3/4	75 %
Jika membeli susu, maka membeli kopi	3/6	50 %
Jika membeli susu, maka membeli roti	3/6	50 %
Jika membeli roti, maka membeli susu	3/4	75 %

CONTOH LAIN:

Pelanggan ke-	Item belanja yg dibeli
1	Broccoli, green peppers, corn
2	Asparagus, squash, corn
3	Corn, tomatoes, beans, squash
4	Green peppers, corn, tomatoes, beans
5	Beans, asparagus, broccoli
6	Squash, asparagus, beans, tomatoes
7	Tomatoes, corn
8	Broccoli, tomatoes, green peppers
9	Squah, asparagus, beans
10	Beans, corn
11	Green peppers, broccoli, beans, squash
12	Asparagus, beans, squash
13	Squash, corn, asparagus, beans
14	Corn, green peppers, tomatoes, beans, brocolli

Misalnya: $\Phi = 4$, Support = 30 %, Confidence = 70 %

Dari Itemset Sering	Aturan asosiasi	Support		Confidence	
{Asparagus, Beans}	Jika beli asparagus, maka beli beans				
	Jika beli beans, maka beli asparagus				
{Asparagus, Squash}	Jika beli asparagus, maka beli squash				
	Jika beli squash, maka beli asparagus				
{Beans, Corn}	Jika beli beans, maka beli corn				
	Jika beli corn, maka beli beans				
{Beans, Squash}	Jika beli beans, maka beli squash				
	Jika beli squash, maka beli beans				

Dari Itemset Sering	Aturan asosiasi	Support		Confidence	
{Beans, Tomatoes}	Jika beli beans, maka beli tomatoes				
	Jika beli tomatoes, maka beli beans				
{Broccoli, Green peppers}	Jika beli broccoli, maka beli green peppers				
	Jika beli green peppers, maka beli broccoli				
{Corn, Tomatoes}	Jika beli corn, maka beli tomatoes				
	Jika beli tomatoes, maka beli corn				
{Asparagus, Beans, Squash}	Jika beli asparagus dan beans, maka beli squash				
	Jika beli asparagus dan squash, maka beli beans				
	Jika beli beans dan squash, maka beli asparagus				

Misalnya: Φ = 4, Support = 30 %, Confidence = 70 %

Dari Itemset Sering	Aturan asosiasi	Support		Confidence	
{Asparagus, Beans}	Jika beli asparagus, maka beli beans	5/14	35,7 %	5/6	83,3 %
	Jika beli beans, maka beli asparagus	5/14	35,7 %	5/10	50,0 %
{Asparagus, Squash}	Jika beli asparagus, maka beli squash	5/14	35,7 %	5/6	83,3 %
	Jika beli squash, maka beli asparagus	5/14	35,7 %	5/7	71,4 %
{Beans, Corn}	Jika beli beans, maka beli corn	5/14	35,7 %	5/10	50,0 %
	Jika beli corn, maka beli beans	5/14	35,7 %	5/8	62,5 %
{Beans, Squash}	Jika beli beans, maka beli squash	6/14	42,9 %	6/10	60,0 %
	Jika beli squash, maka beli beans	6/14	42,9 %	6/7	85,7 %

Dari Itemset Sering	Aturan asosiasi	Support		Confidence	
{Beans, Tomatoes}	Jika beli beans, maka beli tomatoes	4/14	28,6 %	4/10	40,0 %
	Jika beli tomatoes, maka beli beans	4/14	28,6 %	4/6	66,7 %
{Broccoli, Green peppers}	Jika beli broccoli, maka beli green peppers	4/14	28,6 %	4/5	80,0 %
	Jika beli green peppers, maka beli broccoli	4/14	28,6 %	4/5	80,0 %
{Corn, Tomatoes}	Jika beli corn, maka beli tomatoes	4/14	28,6 %	4/8	50,0 %
	Jika beli tomatoes, maka beli corn	4/14	28,6 %	4/6	66,7 %
{Asparagus, Beans, Squash}	Jika beli asparagus dan beans, maka beli squash	4/14	28,6 %	4/5	80,0 %
	Jika beli asparagus dan squash, maka beli beans	4/14	28,6 %	4/5	80,0 %
	Jika beli beans dan squash, maka beli asparagus	4/14	28,6 %	4/6	66,7 %

Hasilnya :

$\Phi = 4$, Min. Support = 30 %, Min. Confidence = 70 %

Dari Itemset Sering	Aturan asosiasi	Support		Confidence	
{Asparagus, Beans}	Jika beli asparagus, maka beli beans	5/14	35,7 %	5/6	83,3 %
{Asparagus, Squash}	Jika beli asparagus, maka beli squash	5/14	35,7 %	5/6	83,3 %
	Jika beli squash, maka beli asparagus	5/14	35,7 %	5/7	71,4 %
{Beans, Corn}	Jika beli corn, maka beli beans	5/14	35,7 %	5/8	62,5 %
{Beans, Squash}	Jika beli beans, maka beli squash	6/14	42,9 %	6/10	60,0 %
	Jika beli squash, maka beli beans	6/14	42,9 %	6/7	85,7 %

FP-Growth

Rule Assosiasi

- Pertama kali di kenalkan oleh R. Agrawal pada tahun 1993.
- Merupakan suatu teknik dalam data mining untuk menentukan aturan asosiatif antara suatu kombinasi item.
- Sehingga menghasilkan suatu rule untuk memprediksi kombinasi yang sama akan terjadi kembali.
- Pertama kali digunakan untuk analisa pola belanja *customer*.

Algoritma Apriori

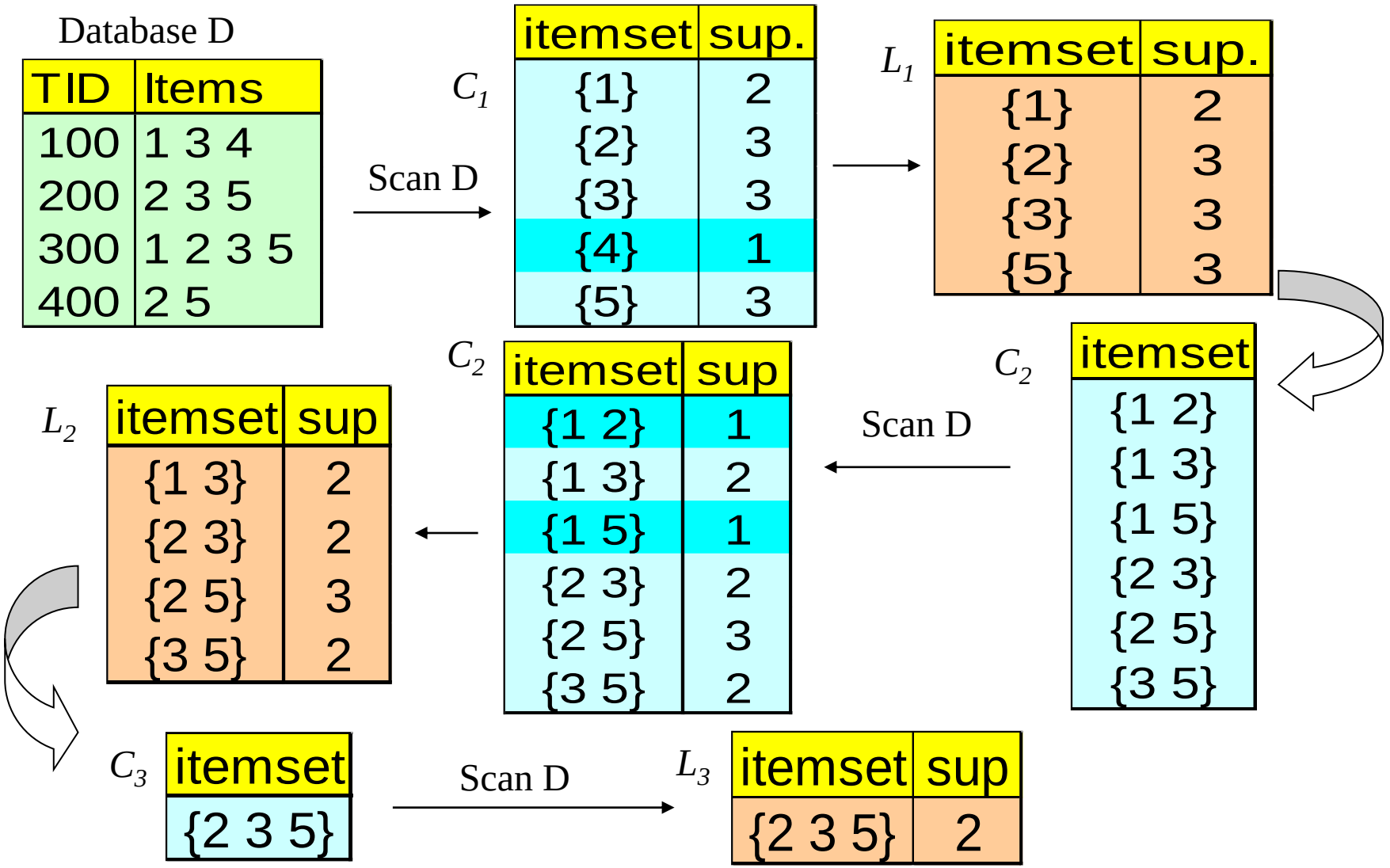
- Apriori adalah algoritma untuk mendapatkan *frequent item set* dan rule asosiasi terhadap transaksi dalam database.
- Di proses dengan cara mengidentifikasi setiap set item yang sering muncul, dan memperluasnya ke kumpulan item yang lebih besar selagi item tersebut cukup sering muncul.

Bagaimana dengan performa Apriori?

Algoritma Apriori

1. Pilih frequent (k-1)-itemsets untuk menghasilkan kandidat k-itemsets
2. Scan database untuk menentukan frequent k-itemsets berikutnya.

Contoh Apriori



Apriori

- Tidak efisien untuk mengelola jumlah frequent itemset yang besar, Karena melakukan scaning secara berulang-ulang untuk menentukan *frequent itemset*.
- Apabila jumlah frequent *1-itemsets* sebanyak 10^4, maka algoritma Apriori akan membutuhkan lebih dari 10^7 *2-itemsets* untuk membuat itemset berikutnya.

Apriori

- Untuk menghasilkan 100-itemset
- Diperlukan sebanyak $2^{100}-1$ kandidat yang harus dibuat

$$2^{100}-1=1.27*10^{30}$$

(Seberapa besar angka ini?)

- $7*10^{27} \approx$ jumlah atom dalam tubuh manusia
- $6*10^{49} \approx$ jumlah atom untuk bumi
- $10^{78} \approx$ jumlah atom untuk alam semesta

Metode-metode Rule Assosiasi selain Apriori

- ECLAT
- FP-Growth
- AprioriDP
- Context Based Association Rule Mining Algorithm
- Node-set-based algorithms
- GUHA procedure ASSOC
- OPUS searc

Definisi FP-Growth

- Salah satu alternatif algoritma untuk menentukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data.
- Salah satu teknik yang terukur.
- Pengembangan dari algoritma Apriori.
- Tidak melakukan *candidate generation* dalam proses pencarian frequent itemset.
- Informasi frequent itemset disimpan dalam bentuk struktur pohon, biasanya disebut *FP-Tree*.
- Hanya diperlukan “2x scan” terhadap database
  1. Buat struktur data berupa pohon.
  2. Ekstrak frequent item set langsung dari pohon yang sudah dibuat.

Contoh FP tree

Suppose we have the following DataBase:

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Step 1 - Calculate Minimum support

- First should calculate the minimum support count. Question says minimum support should be 30%. It calculate as follows:

  Minimum support count(30/100 * 8) = 2.4

  As a result, 2.4 appears but to empower the easy calculation it can be rounded to to the ceiling value. Now, Minimum support count is ceiling(30/100 * 8) = 3

Step 2 -Find frequency of occurrence

Mencari frekuensi kemunculan setiap item dalam tabel Database. Misalnya, item A muncul di baris 1, baris 2, baris 3, baris 4, dan baris 7. Totalnya 5 kali muncul di tabel Database. Anda dapat melihat penghitungan frekuensi kemunculan setiap item pada Tabel

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Table 1 – Snapshot of the Database

TID	Frequency
A	5
B	6
C	3
D	6
E	4

Table 2 – Frequency of Occurance

Step 3 -Prioritize the items

Pada Tabel 2 Anda dapat melihat angka-angka yang ditulis dengan pena Merah. Itulah prioritas setiap item menurut frekuensi kemunculannya. Item B mendapat prioritas tertinggi (1) karena jumlah kemunculannya paling banyak. Pada saat yang sama Anda mempunyai kesempatan untuk menjatuhkan item yang tidak memenuhi persyaratan dukungan minimum. Misalnya jika Database berisi F yang memiliki frekuensi 1, maka Anda dapat membuangnya.

TID	Frequency	Priority
A	5	3
B	6	1
C	3	5
D	6	2
E	4	4

Table 2 – Frequency of Occurence

Step 4 -Order the items according to priority

Seperti terlihat pada Tabel 3, kolom baru ditambahkan pada Tabel 1. Pada kolom Barang Pesanan semua barang diantri sesuai prioritasnya, yang disebutkan dalam tinta Merah pada Tabel 2. Misalnya pada kasus pemesanan baris 1 , item dengan prioritas tertinggi adalah B dan setelah itu masing-masing D, A dan E.

TID	Items	Ordered Items
1	E, A, D, B	B, D, A, E
2	D, A, C, E, B	B, D, A, E, C
3	C, A, B, E	B, A, E, C
4	B, A, D	B, D, A
5	D	D
6	D, B	B, D
7	A, D, E	D, A, E
8	B, C	B, C

Table 3 – New version of the Table 1

Step 5 -Order the items according to priority

- Sebagai hasil dari langkah sebelumnya, didapatkan tabel barang pesanan (Tabel 3). Sekarang saatnya menggambar pohon FP. Sebutkan baris demi baris
- Baris 1: Perhatikan bahwa semua pohon FP memiliki simpul 'null' sebagai simpul akar. Jadi gambarlah simpul akarnya terlebih dahulu dan tempelkan masing-masing item pada baris 1 satu per satu. (Lihat Gambar 1) Dan tuliskan kejadiannya di depannya.

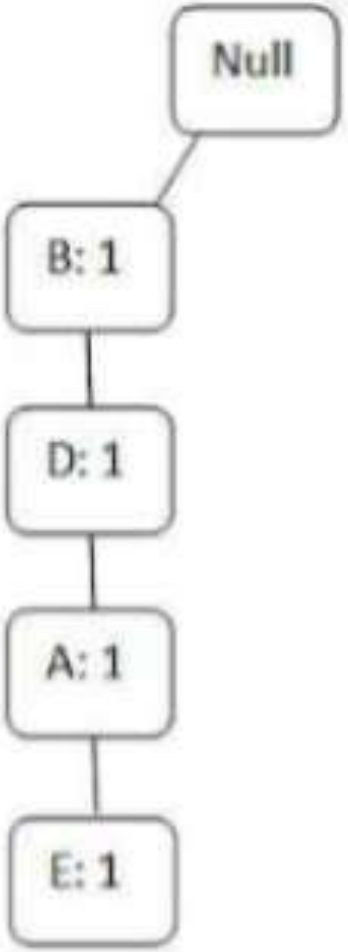


Figure 1-FP tree for Row 1

- Baris 2: Kemudian perbarui pohon di atas (Gambar 1) dengan memasukkan item pada baris 2. Item pada baris 2 adalah B,D,A,E,C. Kemudian tanpa membuat cabang lain Anda dapat melewati cabang sebelumnya hingga E dan kemudian Anda harus membuat simpul baru setelah itu untuk C. Kasus ini sama dengan skenario perjalanan melalui jalan raya untuk mengunjungi kota-kota di negara tersebut. Anda harus melalui jalan yang sama untuk mencapai kota lain yang dekat dengan kota tersebut. Saat Anda melewati cabang untuk kedua kalinya, Anda harus menghapus satu dan menulis dua untuk menunjukkan dua kali Anda mengunjungi node tersebut. Jika Anda mengunjungi tiga kali maka tulislah tiga setelah menghapus dua

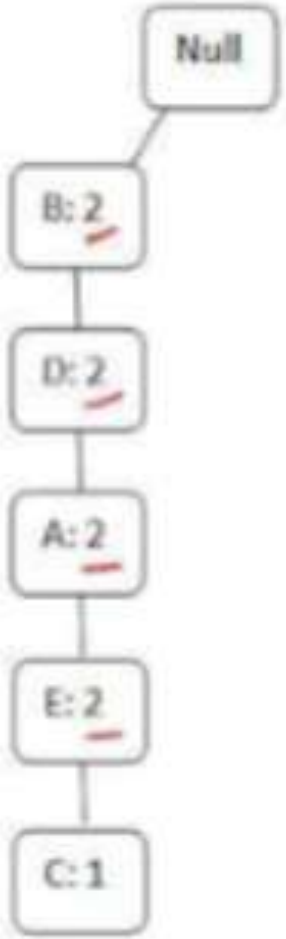


Figure 2-FP tree for Row 1,2

- Baris 3: Di baris 3 Anda harus mengunjungi B,A,E dan C masing-masing. Jadi Anda mungkin berpikir Anda bisa mengikuti cabang yang sama lagi dengan mengganti nilai B,A,E dan C Tapi Anda tidak bisa melakukan itu, Anda punya kesempatan untuk melewati B. Tapi Anda tidak bisa menghubungkan B ke A yang ada dan menyalip D. Sebagai hasilnya, Anda harus menarik A lagi dan menghubungkannya ke B lalu menghubungkan E baru ke A itu dan C baru ke E baru.

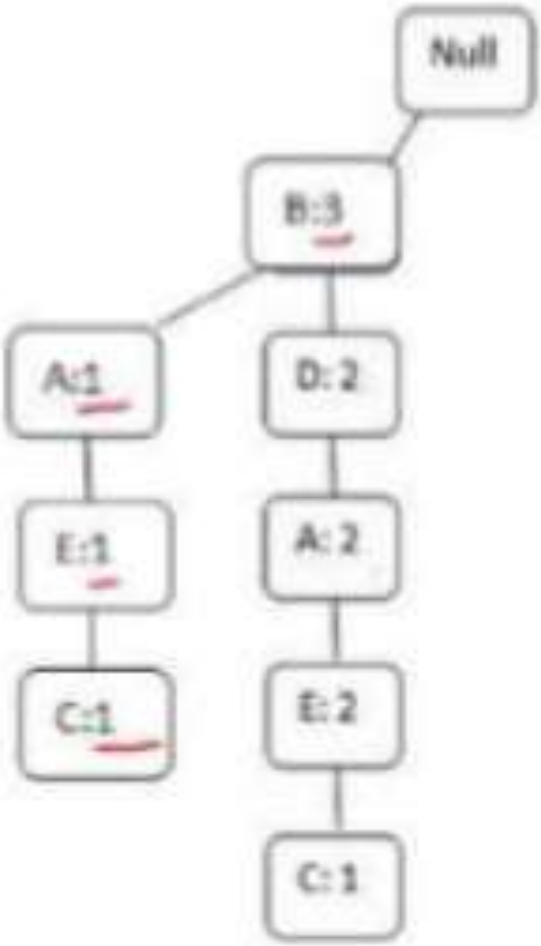


Figure 3-After adding 3rd row

- Baris 4: Kemudian baris 4 berisi B,D,A. Sekarang kita tinggal mengganti nama frekuensi kemunculan di cabang yang ada. Seperti B:4,D,A:3.
- Baris 5: n mentah kelima hanya memiliki item D. Sekarang kita memiliki kesempatan untuk menarik cabang baru dari node 'null'. Lihat Gambar 4.

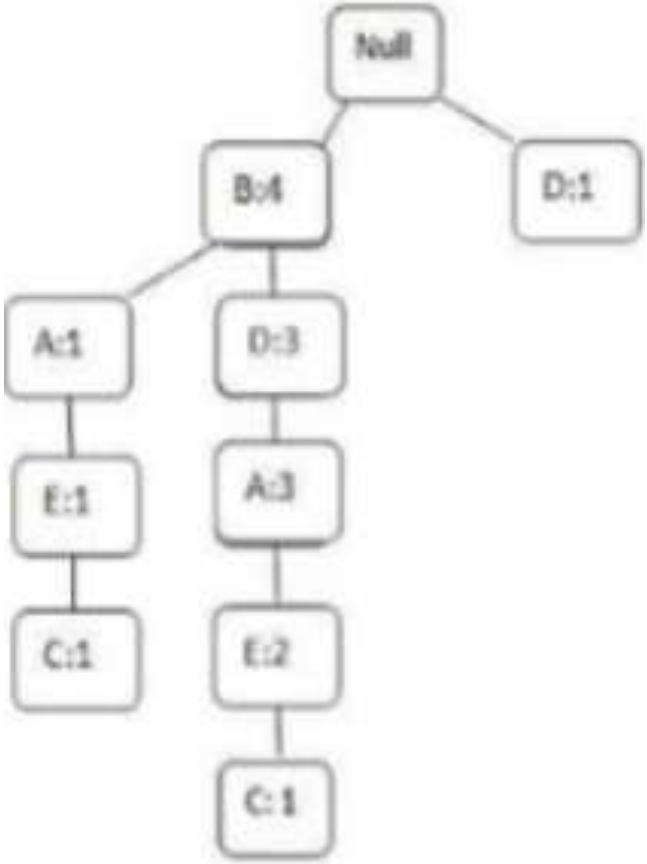


Figure 4-Connect D to null node

- Baris 6: B dan D muncul di baris 6. Jadi ubah saja B:4 menjadi B:5 dan D:3 menjadi D:4.
- Baris 7: Lampirkan dua node baru A dan E ke node D yang tergantung pada node nol. Kemudian tandai D,A,E sebagai D:2,A:1 dan E:1.
- Baris 8 :(baris terakhir) Pasang simpul baru C ke B. Ubah waktu lintasan.(B:6,C:1)

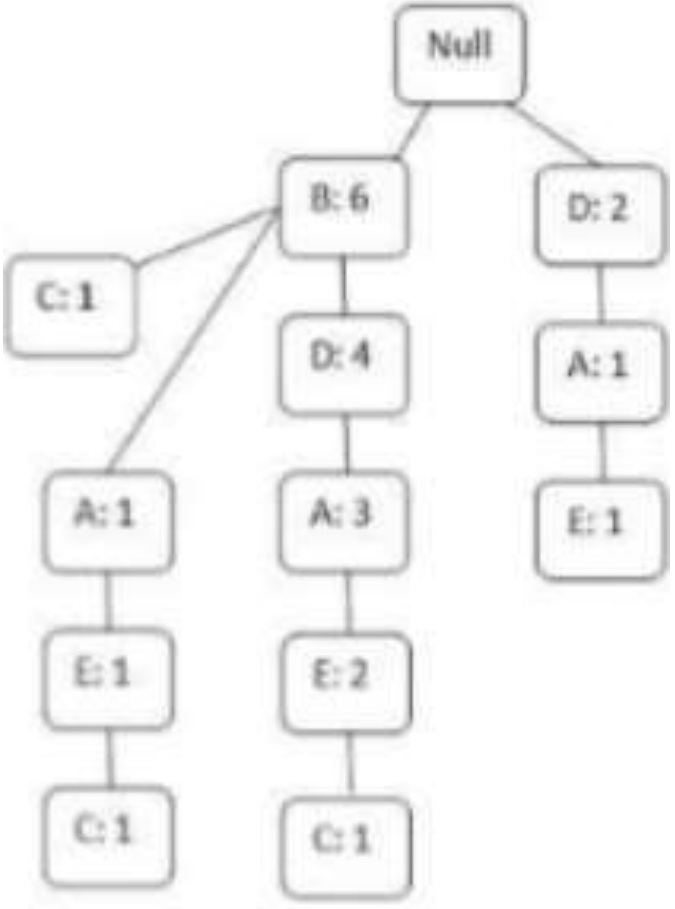


Figure 5-Final FP tree

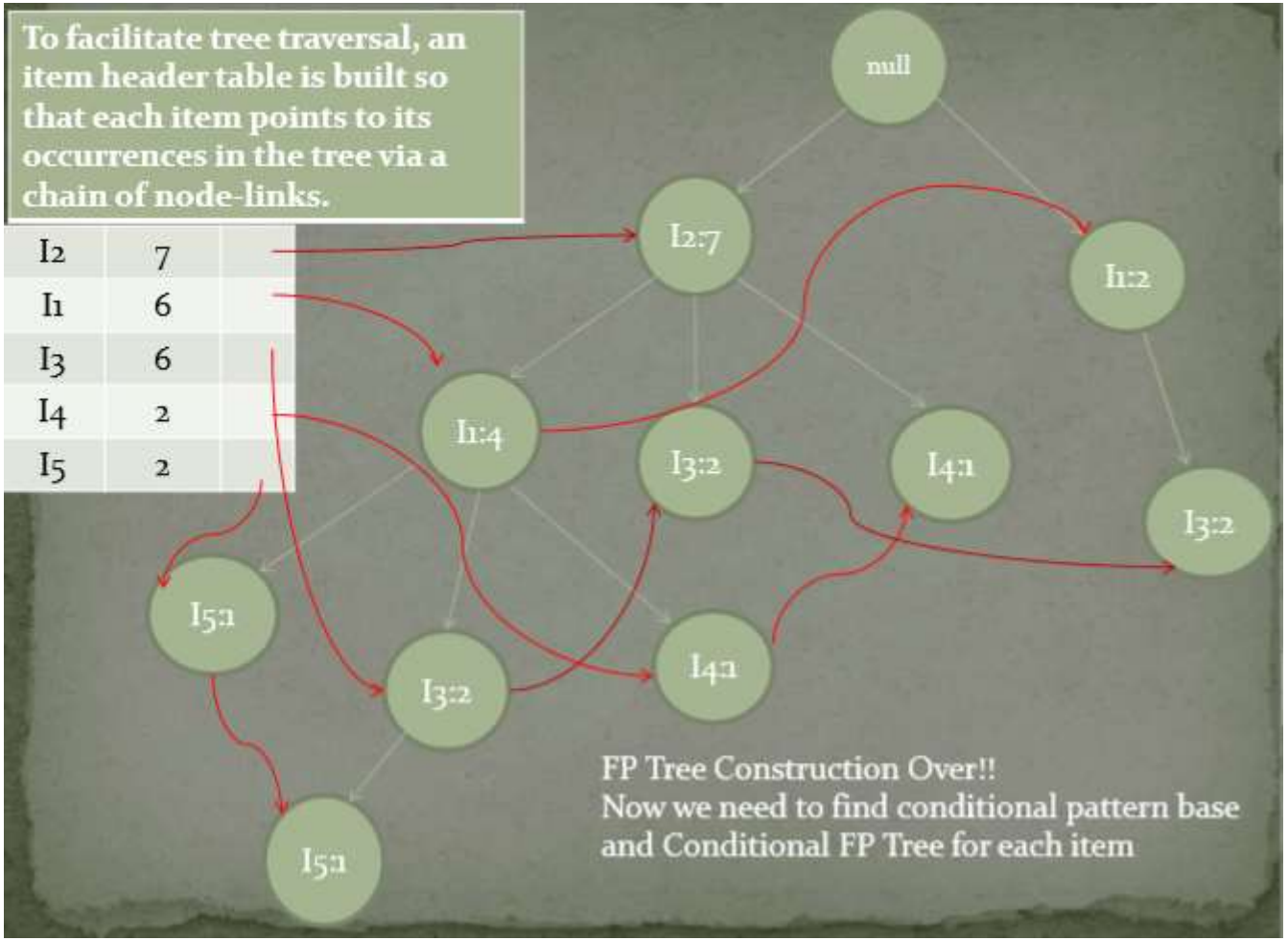
Step 6 -Validation

Setelah lima langkah terakhir pohon FP sebagai berikut: Gambar 5.

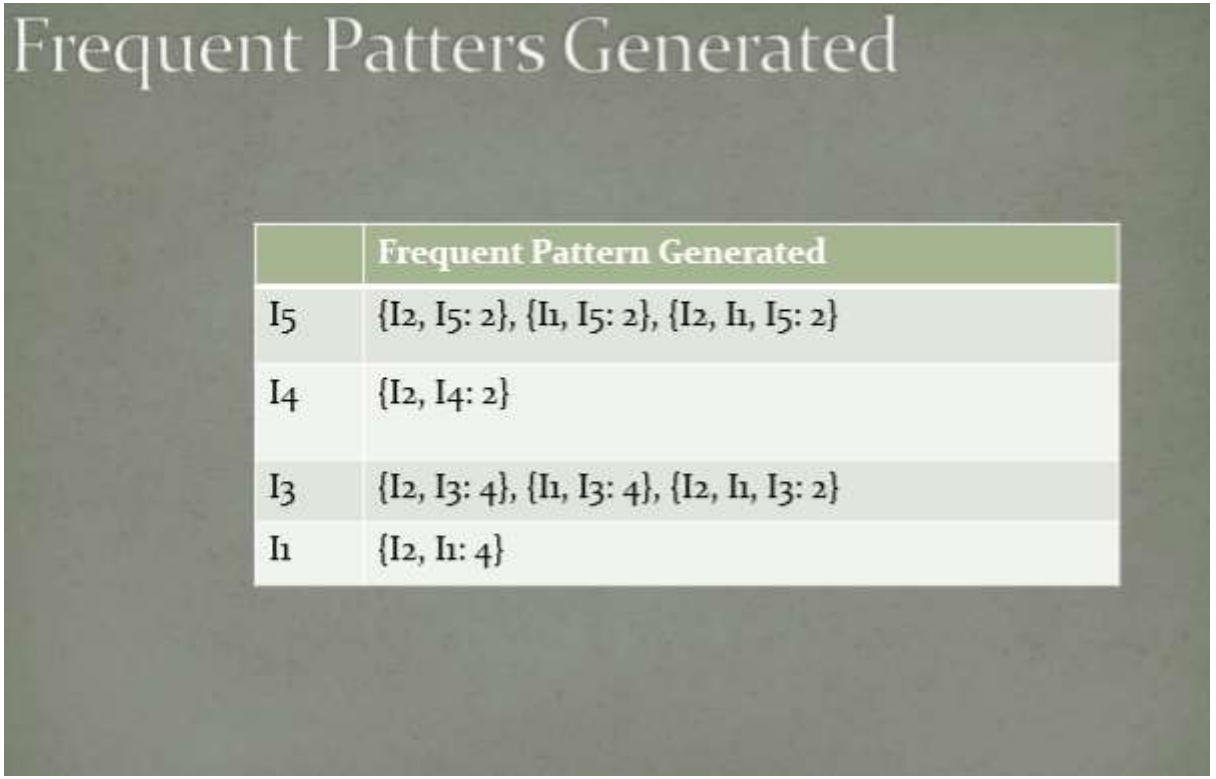
Bagaimana kita tahu apakah ini benar?

Sekarang hitung frekuensi kemunculan setiap item pada pohon FP dan bandingkan dengan Tabel 2. Jika keduanya dihitung sama, maka ini adalah titik positif yang menunjukkan bahwa pohon Anda benar.

Step 6 -Validation



Frequent Patterns Generated



Kesimpulan

- Struktur data *frequent itemset* tersimpan lebih ringkas, sehingga penggunaan memori komputer lebih sedikit, dan proses pencarian *frequent itemset* menjadi lebih cepat.
- Selain lebih ringkas, seluruh informasi *frequent itemset* tersimpan dalam satu pohon.
- Dengan menggunakan algoritma FP Growth, maka pemindaian kumpulan data transaksi hanya dilakukan dua kali, sehinga proses pencarian *frequent itemset* jauh lebih efisien dibandingkan dengan algoritma apriori.

Thank You