Analisis Asosiasi

CS 4333 Data Mining

Imelda Atastina

Analisis Asosiasi

- Adalah sebuah metodologi untuk mencari relasi istimewa/menarik yang tersembunyi dalam himpunan data (*data set*) yang besar
- Relasi yang tersembunyi ini dapat direpresentasikan dalam bentuk aturan asosiasi (*association rules*) atau himpunan barang yang seringkali muncul (*frequent itemset*)

Menambang Aturan Asosiasi

- Berdasarkan data set transaki, akan dicari aturan yang dapat memprediksi kejadian bersama sebuah item, berdasarkan kejadian bersama dari item-item lainnya dalam transaksi

Market-Basket transactions

TID	*Items*
1	Roti, Susu
2	Roti, Diaper, Bir, Telur
3	Susu, Diaper, Bir, Coke
4	Roti, Susu, Diaper, Bir
5	Roti, Susu, Diaper, Coke

Contoh Aturan Asosiasi

{Diaper} $\rightarrow$ {Bir},
{Susu, Roti} $\rightarrow$ {Telur,Coke},
{Bir, Roti} $\rightarrow$ {Susu},

Tanda implikasi diatas berarti kejadian bersama, bukan sebab akibat!

Beberapa Istilah

- Itemset : Koleksi dari sejumlah (satu/lebih)item
  - Contoh: {Bir} , { Susu,Roti, Diaper}
  - k-itemset
    - Item set yang terdiri dari k item
    - Contoh : 3 – item set = { Susu,Roti, Diaper}
- **Support count (σ)**
  - Frekuensi terjadinya sebuah itemset dalam data set
  - Contoh : σ({Milk, Bread,Diaper}) = 2
- **Support (s)**
  - Perbandingan terjadinya sebuah itemset terhadap jumlah seluruh itemset dalam dataset
  - E.g. s({Milk, Bread, Diaper}) = 2/5

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Beberapa Istilah (2)

- **Frequent Itemset**
  - Itemset yang nilai supportnya lebih besar atau sama dengan ” *minsup* threshold” Support Count
- **Associaton Rule**

  adalah ekspresi implikasi (X ->Y), dimana X dan Y adalah itemset yang saling disjoint

  contoh : {Milk, Diaper} → {Beer}

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Parameter Pengevaluasi Aturan

- Support (s)
  - Perbandingan transaksi-transaksi yang mengandung X dan Y
- Confidence (c)
  - Menunjukkan kekerapan munculnya item-item dalam Y pada transaksi yang mengandung X

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Contoh

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

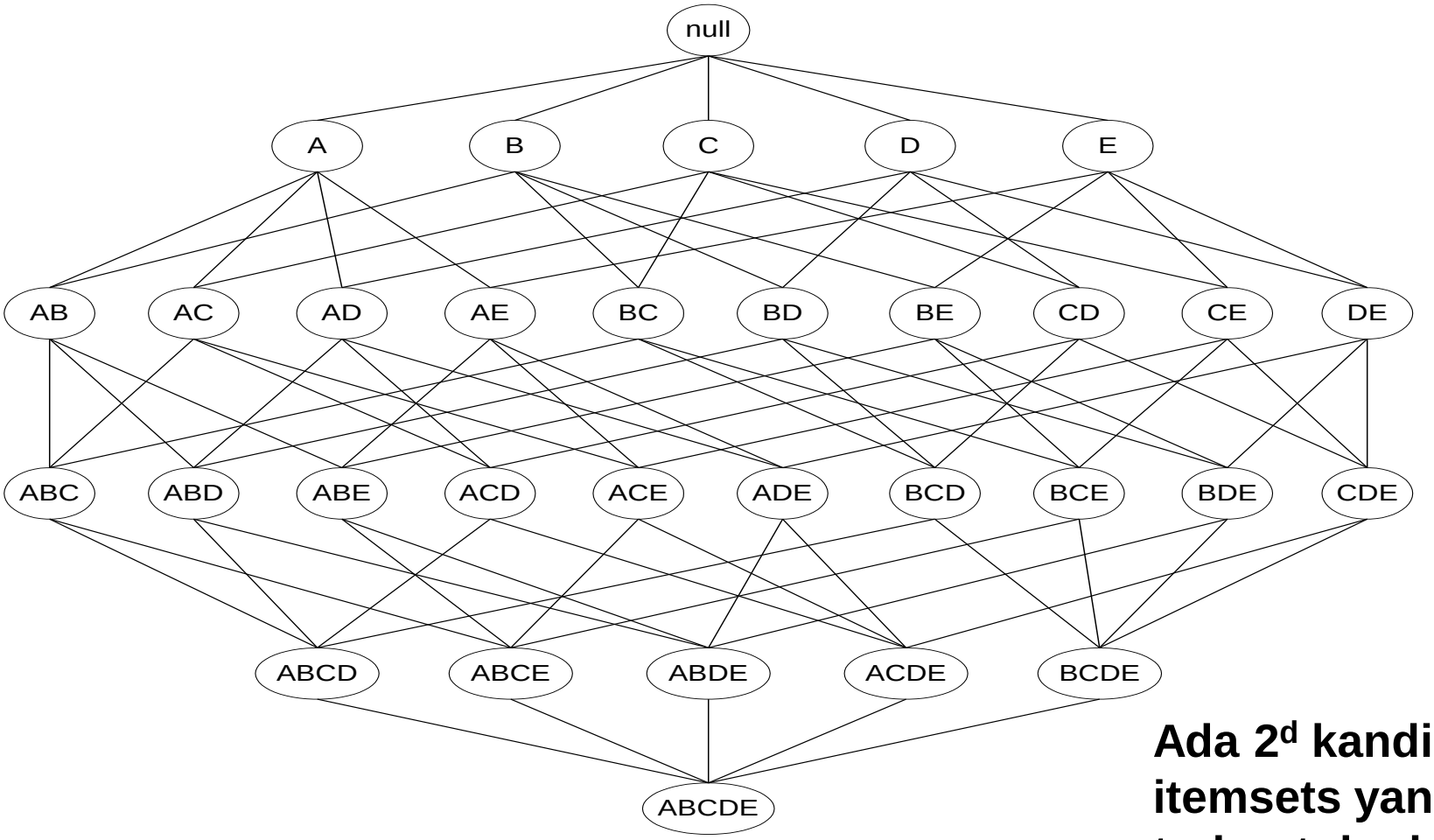
$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Strategi Algoritma Analisis Asosiasi

- Ada 2 langkah besar yang diambil, yaitu :
  1. Frequent Itemset Generation
     - Mengoleksi semua itemset yang memenuhi syarat support ≥ minsup. Itemset-itemset ini disebut *frequent itemset*
  2. Rule Generation
     - Bertujuan membentuk aturan dengan nilai confidence yang tingi dari *frequent itemset* yang telah diperoleh sebelumnya. Aturan ini disebut *strong rules*
- Mengenerate frequent itemset merupakan tahapan yang berat dari sudut pandang komputasi!!!

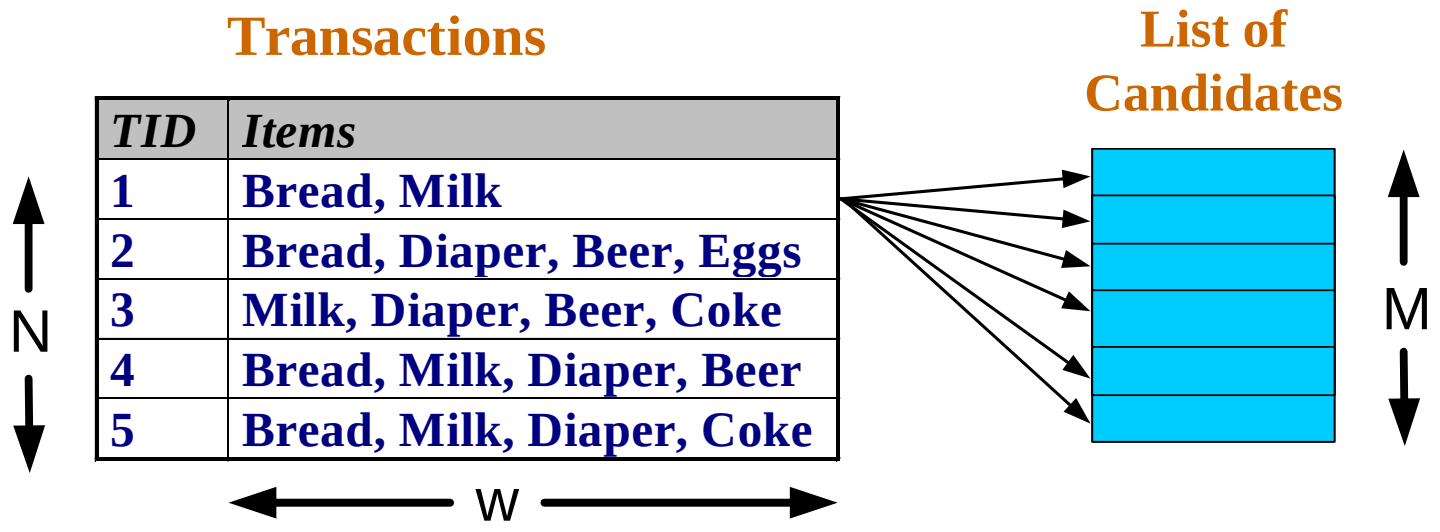
Frequent Itemset Generation



Ada 2^d kandidat itemsets yang terbentuk; d= # item

Frequent Itemset Generation

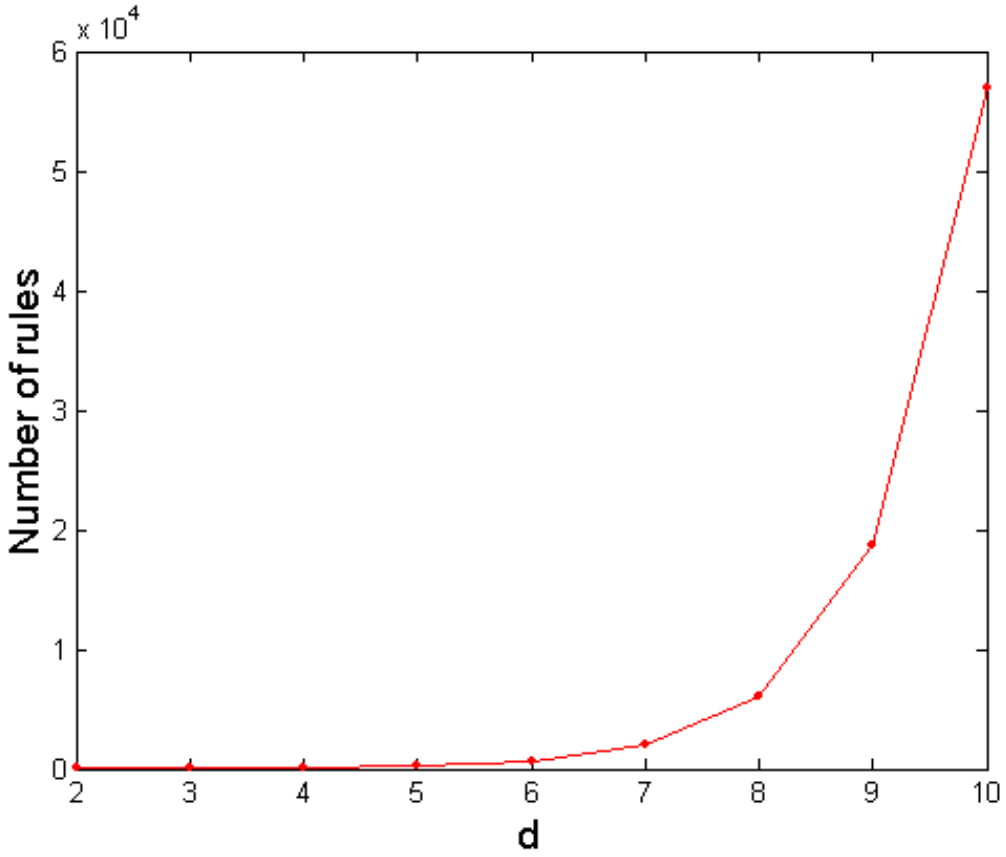
- Brute-force approach:
  - Setiap itemset dalam jaring adalah candidate frequent itemset
  - Hitung support dari setiap kandidat dengan scanning database



  - Bandingkan setiap transaksi terhadap setiap kandidat
  - Kompleksitas ~ O(NMw) => Expensive since $M = 2^d$!!!

Kompleksitas Komputasional

- Jika terdapat d item yang berbeda, maka:
  - Total itemsets = 2^d
  - Total association rules yang mungkin :



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

Jika d=6, R = 602 rules

Strategi Pembentukan Frequent Itemset

- Mereduksi jumlah kandidat (M)
  - Gunakan prinsip Apriori
- Mereduksi jumlah perbandingan (NM)
  - Gunakan struktur data yang efisien untuk menyimpan kandidat atau transaksi
  - Tidak perlu membandingkan semua kandidat terhadap setiap transaksi

Mereduksi jumlah kandidat (M)

- Prinsip Apriori : Jika sebuah itemset merupakan frequent itemset maka subsetnya pun merupakan frequent itemset

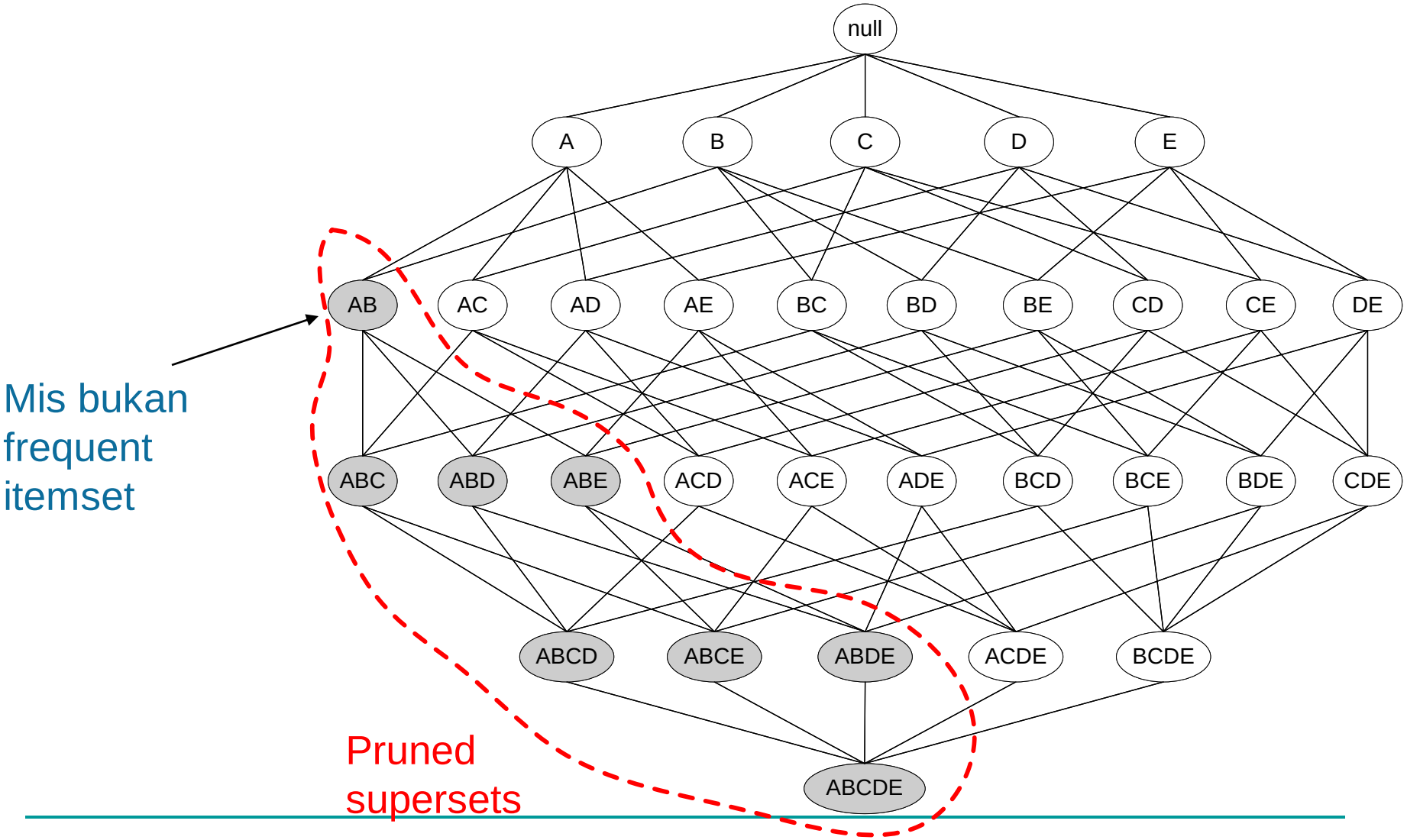
  Contoh : {Susu, Bir, Roti, Diaper} merupakan frequent item set, maka {Susu},{Roti},{Roti, Diaper}, {Susu,Bir,Roti}, dst juga merupakan frequent itemset

- Sifat anti-monotone

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support dari sebuah itemset tidak akan lebih besar dari support subsetnya

Ilustrasi Prinsip Apriori



Ilustrasi Prinsip Apriori (2)

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,
$^{6}C_{1} + {}^{6}C_{2} + {}^{6}C_{3} = 41$
With support-based pruning,
$6 + 6 + 1 = 13$

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

Algoritma Apriori

- Misalkan k=1
- Bentuk frequent itemsets yang terdiri dari k item
- Ulangi hingga tidak ada lagi frequent itemsets yang baru
  - Bentuk kandidat itemset dengan panjang (k+1) dari frequent itemset dengan panjang k
  - Buang kandidat itemsets yang berisi subset dengan panjang k yang tidak frequent
  - Hitung support dari setiap kandidat dengan scannding basisdata
  - Eliminasi kandidat yang infrequent

Pembentukan Rule (1)

- Misalkan ada frequent itemset L, cari subsets yang tidak hampa $f \subset L$ sedemikian sehingga $f \rightarrow L - f$ memenuhi nilai minimum confidence
  - Mis {A,B,C,D} adalah frequent itemset, maka kandidat rules:

$ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$,
$A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$
$AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BC \rightarrow AD$,
$BD \rightarrow AC$, $CD \rightarrow AB$,

- Jk $|L| = k$, maka akan terdapat $2^k - 2$ kandidat association rules (tanpa $L \rightarrow \varnothing$ and $\varnothing \rightarrow L$)

Pembentukan Rule(2)

- Bagaimana membentuk rules dari frequent itemset dengan efisien?
  - Secara umum, confidence tidak bersifat anti-monotone

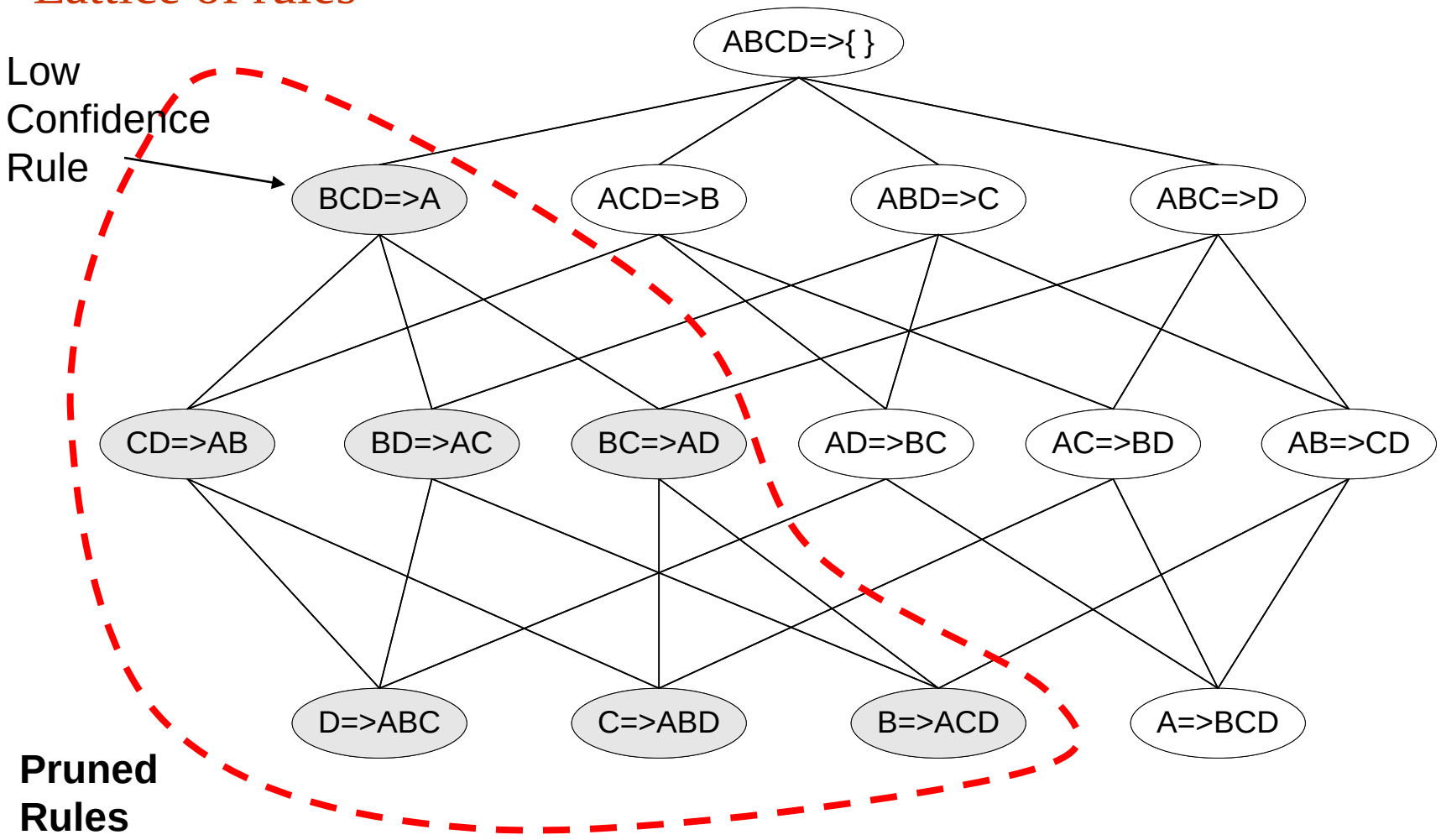
    c(ABC $\rightarrow$D) dapat lebih besar/kecil c(AB $\rightarrow$D)

  - Tetapi nilai confidence dari rules yg berasal dari itemset yang sama bersifat anti-monotone
  - e.g., L = {A,B,C,D}:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

Pembentukan Rule Algoritma Apriori

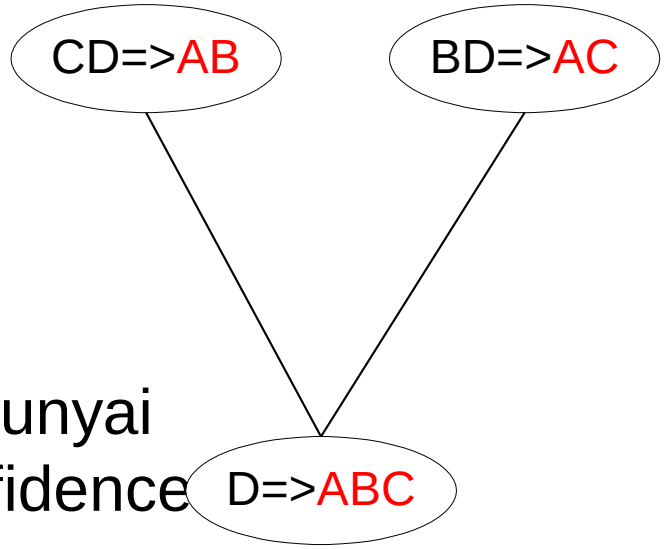
Lattice of rules



Pembentukan Rule Algoritma Apriori

- Kandidat rule dibentuk dengan cara menggabungkan 2 rules yang memiliki prefix yang sama sebagai konsekuennya

- join(CD=>AB,BD=>AC) sehingga terbentuk rule D => ABC



- Buang rule D=>ABC jika ia mempunyai subset AD=>BC dengan nilai confidence

Contoh :

Gunakan algoritma apriori untuk membentuk aturan analisis asosiasi pengklasifikasi dari data pada tabel

Gunakan minimum support = 3

dan min conf = 70%

TID	List of item
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Algoritma FP-Growth

- Gunakan representasi terkompresi basis data dengan memanfaatkan FP-tree

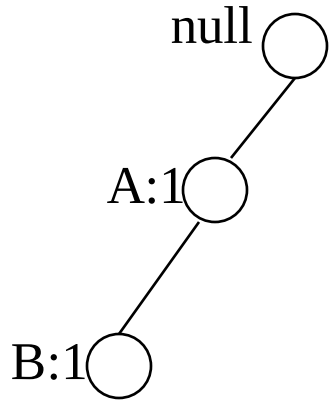
- Setelah FP-tree terbentuk, gunakan teknik divide-and-conquer secara rekursif untuk menambang *frequent itemsets*

Pembentukan FP-tree

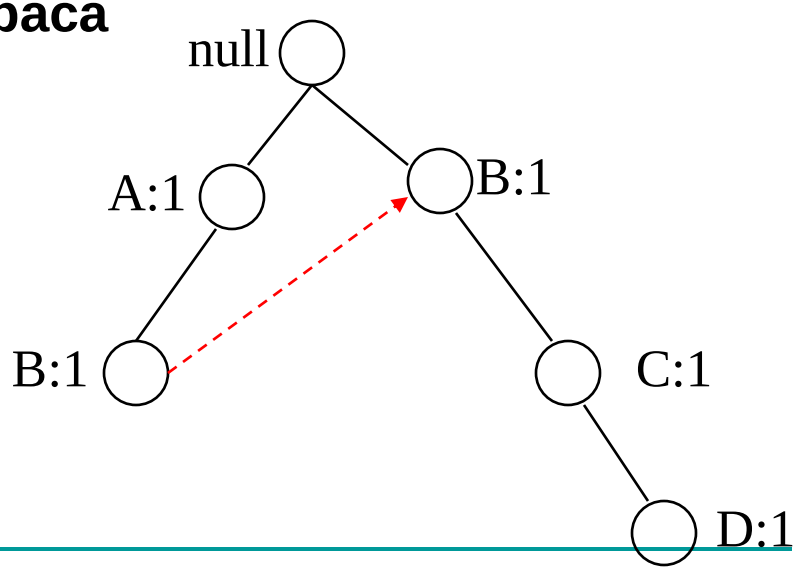
L1 : Susun 1-item dgn nilai support count menurun

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

Setelah membaca TID=1:



Setelah membaca TID=2:



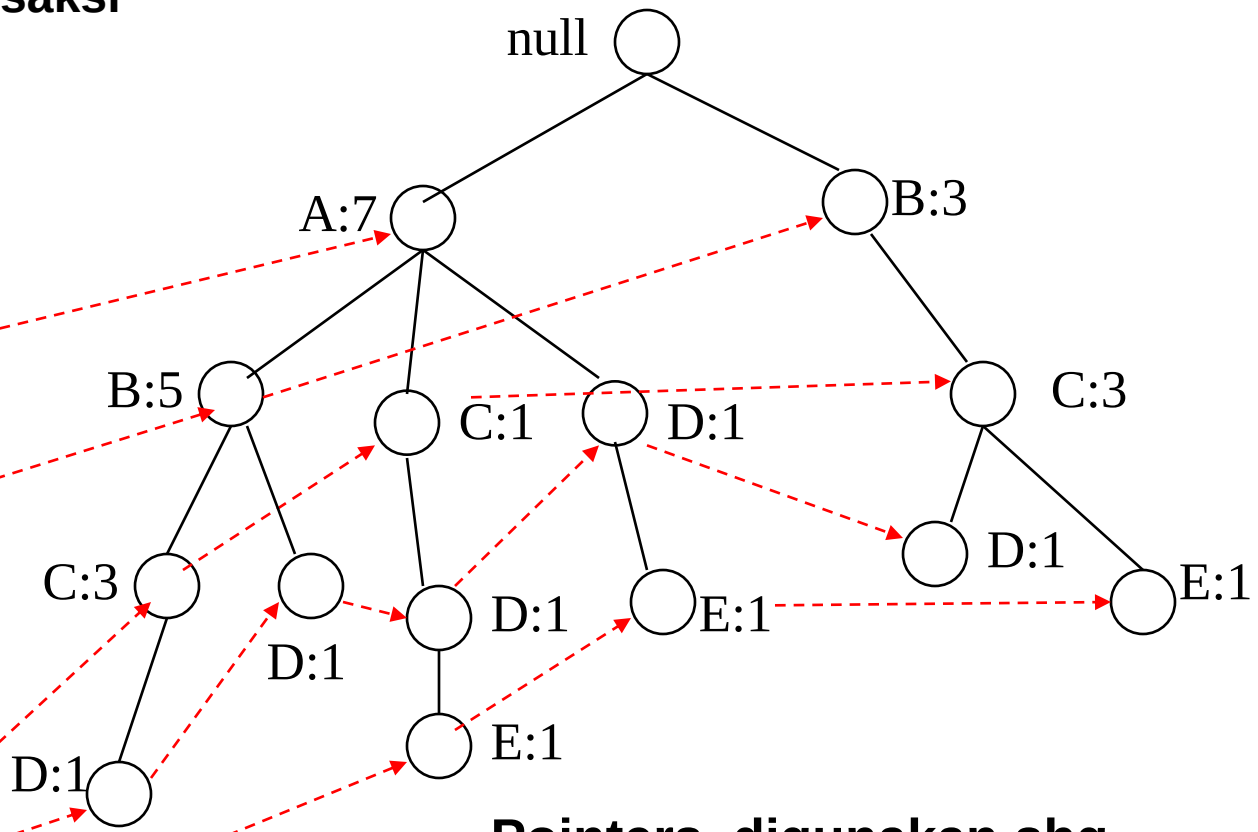
FP-Tree Construction

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

Database Transaksi

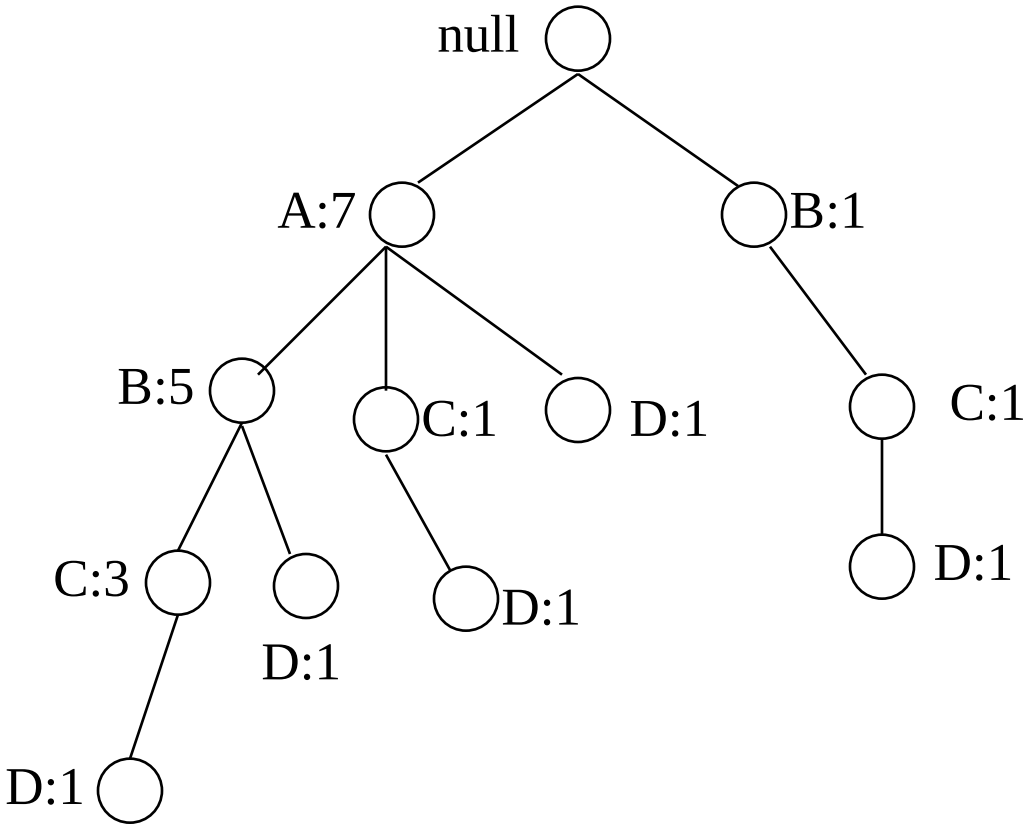
Header table

Item	Pointer
A	
B	
C	
D	
E	



Pointers digunakan sbg bantuan dalam menelusuri pohon frequent pattern

FP-growth



Conditional Pattern base untuk D:

P = {(A:1,B:1,C:1),
(A:1,B:1),
(A:1,C:1),
(A:1),
(B:1,C:1)}

Secara rekursif terapkan proses FP-Growth pada P

Mis minsup=1,maka

Frequent Itemsets yang diperoleh :

AD, BD, CD, ACD,BCD