MoF-DAC
Ministry of Finance
Data Analytics Community

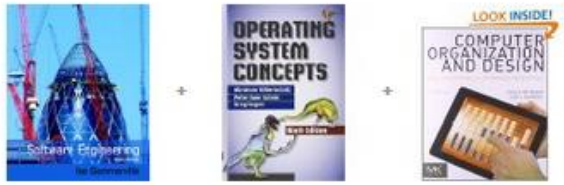
MACHINE LEARNING ASSOCIATION RULES

Ade Satya Wahana

- Pengenalan

Familiar kah?

Frequently Bought Together



Price for all three: $387.88

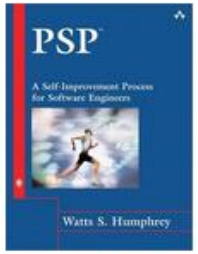
Add all three to Cart

Add all three to Wish List

Some of these items ship sooner than the others. Show details

- ☑ **This item:** Software Engineering (10th Edition) by Ian Sommerville Hardcover $169.67
- ☑ Operating System Concepts by Abraham Silberschatz Hardcover $144.03
- ☑ Computer Organization and Design, Fifth Edition: The Hardware/Software Interface (The Morgan Kaufmann ... by David A. Patterson Paperback $74.18

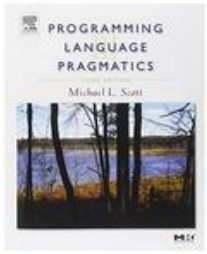
Customers Who Bought This Item Also Bought



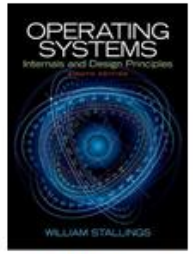
PSP(sm): A Self-Improvement Process for Software Engineers
› Watts S. Humphrey
★★★★☆ 12
Hardcover
$46.41 Prime

Computer Networking: A Top-Down Approach (6th Edition)
› James F. Kurose
★★★★☆ 131
Hardcover
$127.42 Prime

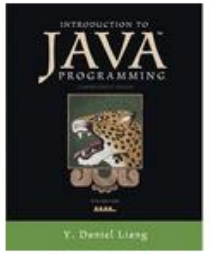
Computer Organization and Design, Fifth Edition: The Hardware/Software
David A. Patterson
★★★½☆ 42
Paperback
$74.18 Prime



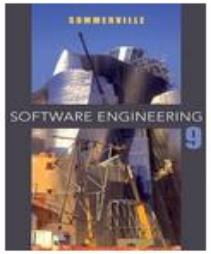
Programming Language Pragmatics, Third Edition
Michael L. Scott
★★★★☆ 24
Paperback
$60.54 Prime



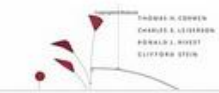
Operating Systems: Internals and Design Principles (8th Edition)
› William Stallings
★★★★½ 10
Hardcover
$141.29 Prime



Introduction to Java Programming, Comprehensive Version (9th Edition)
Y. Daniel Liang
★★★★☆ 82
Paperback



Software Engineering (9th Edition)
Ian Sommerville
★★★☆☆ 29
Hardcover
$140.10 Prime



Show more

Cross-selling

Penjual

Meningkatkan penjualan dg proses penyusunan barang

Pembeli

Kecenderungan membeli produk yang berhubungan

Market-basket analysis
Mempelajari komposisi keranjang belanja dari produk yang dibeli dalam satu kali transaksi

Tantangan:

- Jumlah data yang **sangat banyak**
- **Sparseness** (Setiap keranjang belanja hanya bagian kecil dari keseluruhan transaksi)
- **Heterogeneity** (Setiap pembeli memiliki keunikan tersendiri dalam membeli produk)

Association Rules

- Dikemukakan oleh Agrawal, Imielinski, Swami pada tahun 1993
- Data yang digunakan bersifat kategori
- Algoritma association rule (aturan asosiasi) adalah algoritma yang menemukan atribut yang “muncul bersamaan”
- Association rules digunakan untuk mencari pola, asosiasi, korelasi, atau hubungan sebab akibat dari kumpulan item atau objek dalam suatu transaksi

Use Cases

Penjualan Barang

- * -> Sapu (Apa yang perlu penjual lakukan untuk meningkatkan penjualan sapu)
- Sabun -> * (Stok barang apa yang perlu penjual siapkan untuk mengantisipasi pembeli yang ingin membeli sabun)

Medical Analysis

- Melihat pola antara gejala yang terjadi dengan penyakit
- Melihat hubungan sebab akibat gaya hidup dengan penyakit
- Melihat korelasi antara obat satu dengan obat lainnya

- Association Rules

Important Term

Support
- rate kemunculan barang atau gabungan barang dari keseluruhan transaksi

Confidence
- rate kemunculan kombinasi barang dari jumlah penjualan barang pertama

Strong Association
- Rule yang memenuhi minimum support dan confidence yg diharapkan

Contoh, pada hari kamis malam, **1000 pelanggan** telah **berbelanja** di supermaket ABC, dimana:

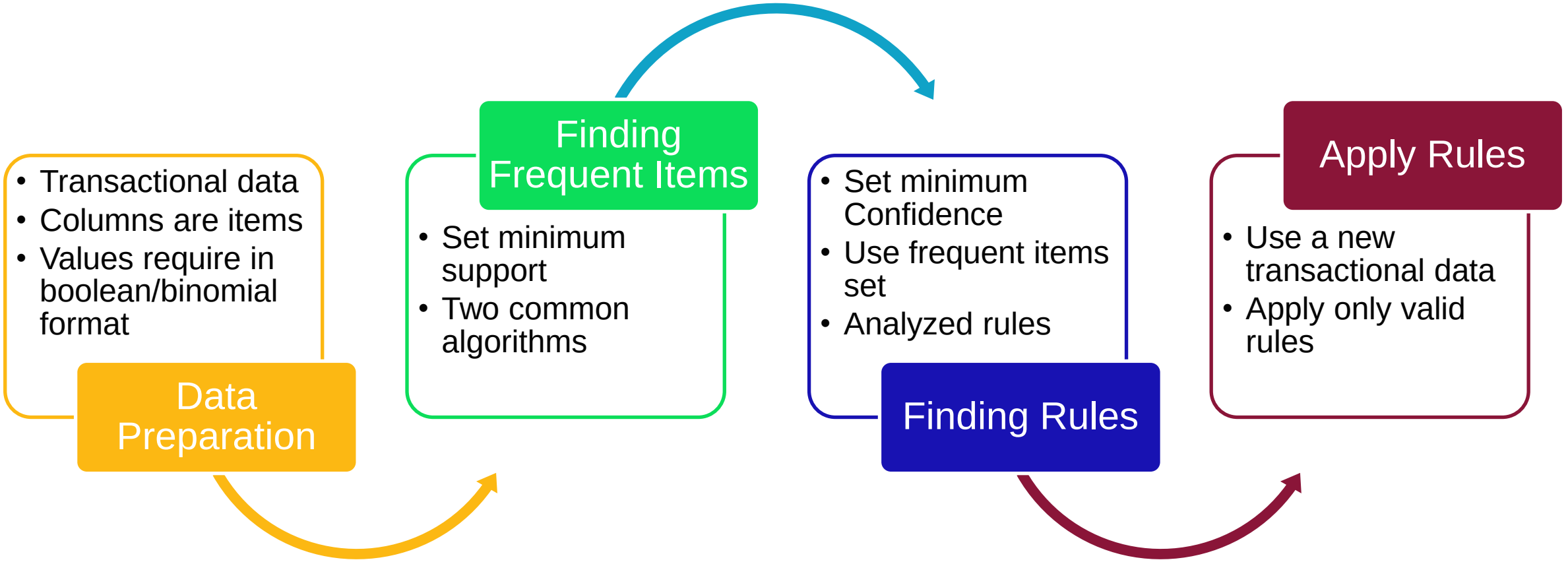
- **200** orang membeli **Sabun Mandi**, dari **200** orang yang membeli **Sabun Mandi**, **50** orangnya membeli **Soda**
- Untuk rule "sabun mandi -> soda", **support = 50/1000 = 5%** dan **confidence = 50/200 = 25%**

Perhitungan Support dan Confidence

$$\text{Support } (A \cap B) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}}$$

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

Association Rules Development Steps



Data Transaksi - pembelajaran dengan metode asosiasi (*FP-Growth*)

ExampleSet (12 examples, 0 special attributes, 10 regular attributes)

Row No.	Gula	Kopi	Aqua	Popok	Sprei	Sabun	Sampo	Kemeja	Celana	Boneka
1	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
2	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0
3	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0
4	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
6	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0
8	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0
9	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
10	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
11	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0

Frequent Itemset

Frequent Itemset

- kumpulan item yang memenuhi minimum **nilai support**

Subset

- **Subset** dari frequent itemset pasti juga merupakan **frequent itemset**
- Jika {A,B} masuk dalam set, baik {A} dan {B} juga masuk

Method

- Apriori
- FP-Growth

Frequent Itemset

ID	Items Pembelian
1	A, C
2	A, B, C
3	A, D
4	B, E, F

Requirement:
Min. Support 50%
Min. Confidence 50%

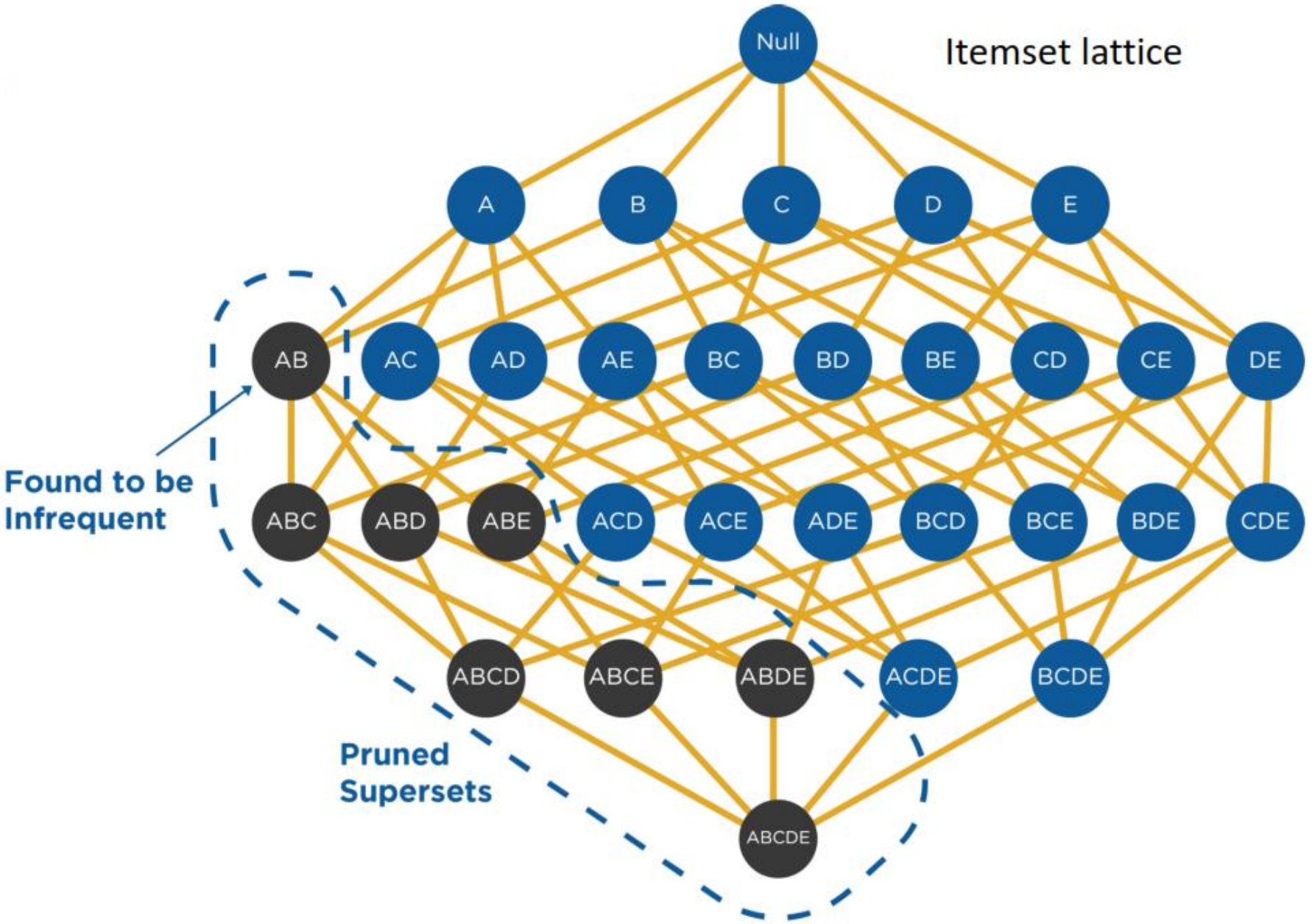
Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A, C}	50%

Mencari Keterhubungan atau Sebab Akibat:

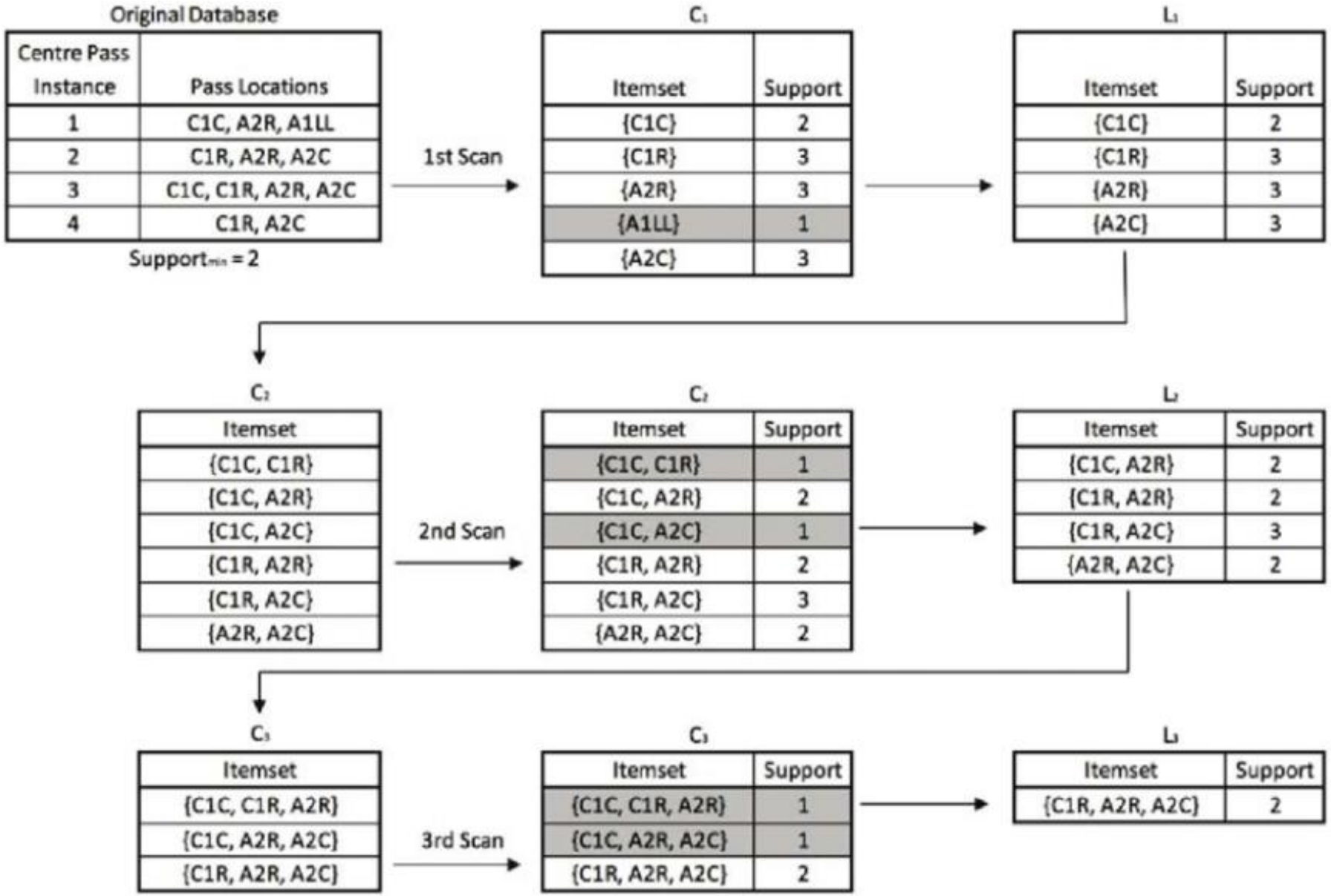
1. Frequent itemset yang memenuhi keterhubungan atau sebab akibat hanya {A, C}
2. Berdasarkan frequent itemset tersebut, perlu dicek apakah memenuhi minimum confidence 50%

Support({A, C}) = 2 / 4 = 50%
Confidence({A, C}) = 50% / 75% = 66.6%

Algoritma Apriori Principle



Algoritma Apriori



Algoritma FP-Growth

Min. Support = 3

2

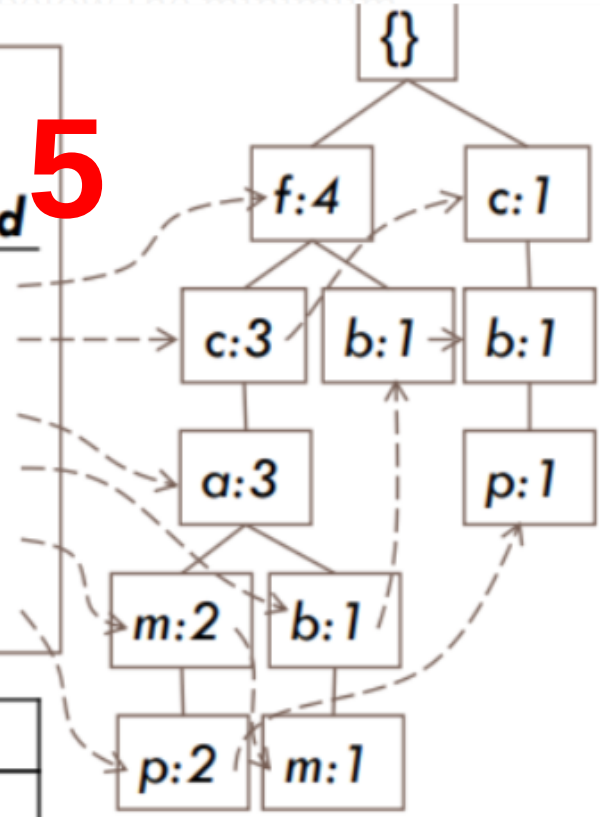
f	4
c	4
a	3
b	3
m	3
p	3

3

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

5



1

TID	Items bought	4 Ordered
100	{a, c, d, f, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, i, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, c, e, f, l, m, n, p}	{f, c, a, m, p}

Frequent itemsets dicari dari ujung tree:
Misal itemsets dg ‘p’:
Hanya {p,c: 3}

Misal itemsets dg ‘m’:
{m,a,c,f: 3} dan subsetnya

Comparison Apriori vs FP-Growth

Parameter	Apriori	FP-Growth
Struktur Storage	Berbentuk Array	Berbentuk Tree
Tipe Searching	Breadth First Search	Divide and Conquer
Teknik	Join dan Prune	Membentuk tree berdasarkan minimum support
Jumlah Scanning	K+1 iterasi	2 iterasi
Penggunaan Memori	Kebutuhan Memori Besar	Kebutuhan Memori Kecil
Waktu Eksekusi	Memakan waktu yang lama	Cepat

Rules Evaluation

Lift Ratio

- ukuran untuk mengetahui kekuatan aturan asosisasi yang telah terbentuk
- digunakan sebagai penentu apakah aturan asosiasi valid atau tidak valid

Insight from Lift

- Lift ratio = 1
  - itemset antiseden dan konsekuen saling independent, tidak ada hubungan kerekatan
- Lift ratio < 1
  - tiap pembelian itemset antiseden mengurangi kemungkinan pembelian konsekuen
- Lift ratio > 1
  - tiap pembelian itemset antiseden meningkatkan kemungkinan pembelian konsekuen

Lift Ratio

Rumus Lift Ratio pada transaksi A dan B

$$\boldsymbol{Lift} = \frac{\boldsymbol{Confidence}(\boldsymbol{A} \cap \boldsymbol{B})}{\boldsymbol{Benchmark\ Confidence}\ (\boldsymbol{A}, \boldsymbol{B})}$$

$$= \frac{\boldsymbol{Confidence}(\boldsymbol{A} \cap \boldsymbol{B})}{\boldsymbol{Support}\ (\boldsymbol{B})}$$

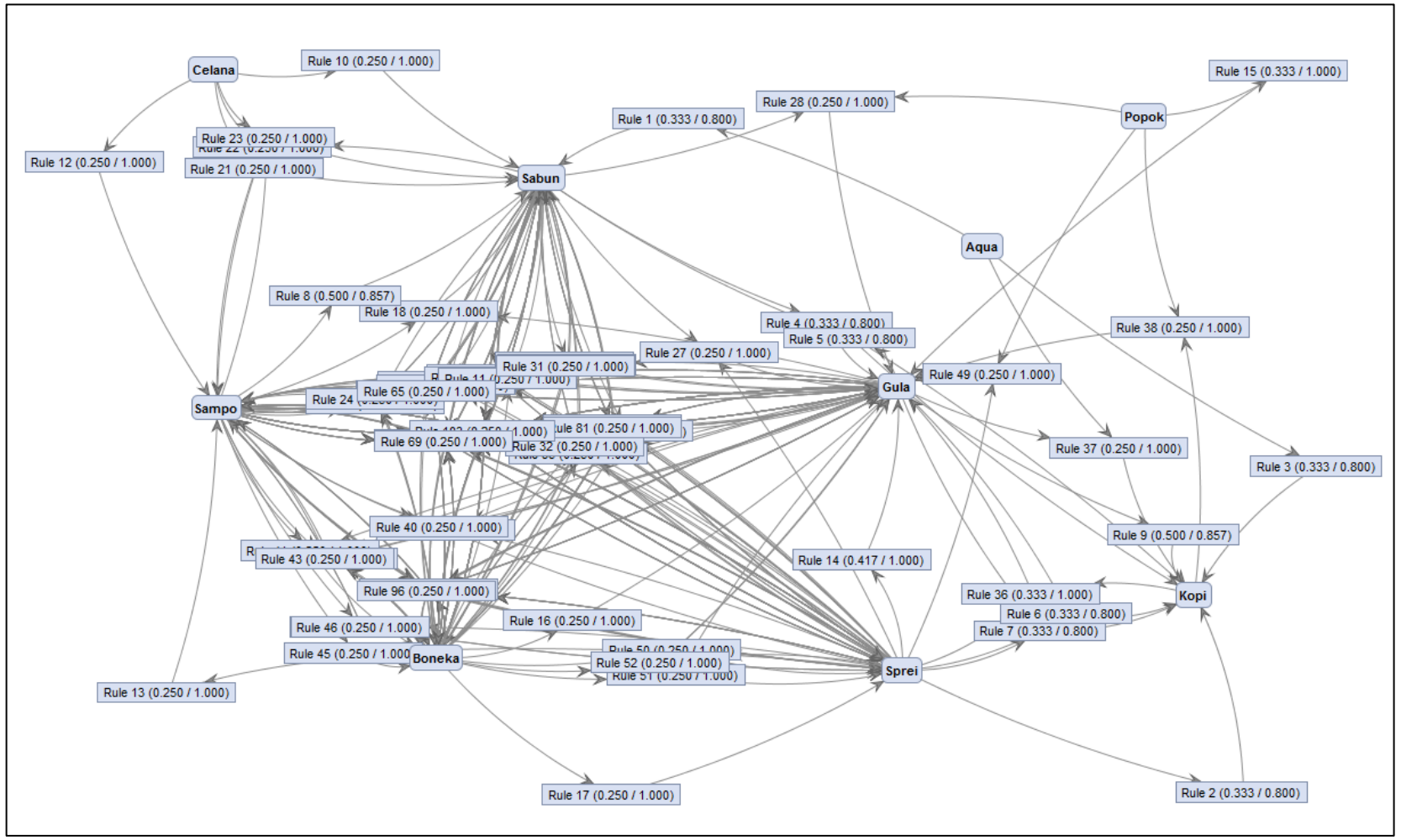
$$\boldsymbol{Benchmark\ Confidence} = \frac{Nc}{N}$$

Nc = jumlah transaksi yang menjadi consequent

N = jumlah transaksi pada basis data

Contoh Penerapan Asosiasi

Data Transaksi – pengetahuan berupa pola asosiasi



AssociationRules

```
Association Rules
[Aqua] --> [Sabun] (confidence: 0.800)
[Sprei] --> [Kopi] (confidence: 0.800)
[Aqua] --> [Kopi] (confidence: 0.800)
[Sabun, Kopi] --> [Gula] (confidence: 0.800)
[Sabun, Gula] --> [Kopi] (confidence: 0.800)
[Sprei] --> [Kopi, Gula] (confidence: 0.800)
[Gula, Sprei] --> [Kopi] (confidence: 0.800)
[Sampo] --> [Sabun] (confidence: 0.857)
[Gula] --> [Kopi] (confidence: 0.857)
[Celana] --> [Sabun] (confidence: 1.000)
[Boneka] --> [Sabun] (confidence: 1.000)
[Celana] --> [Sampo] (confidence: 1.000)
[Boneka] --> [Sampo] (confidence: 1.000)
[Sprei] --> [Gula] (confidence: 1.000)
[Popok] --> [Gula] (confidence: 1.000)
[Boneka] --> [Gula] (confidence: 1.000)
[Boneka] --> [Sprei] (confidence: 1.000)
[Sampo, Gula] --> [Sabun] (confidence: 1.000)
[Sabun, Sprei] --> [Sampo] (confidence: 1.000)
[Sampo, Sprei] --> [Sabun] (confidence: 1.000)
[Celana] --> [Sabun, Sampo] (confidence: 1.000)
[Sabun, Celana] --> [Sampo] (confidence: 1.000)
[Sampo, Celana] --> [Sabun] (confidence: 1.000)
[Boneka] --> [Sabun, Sampo] (confidence: 1.000)
[Sabun, Boneka] --> [Sampo] (confidence: 1.000)
[Sampo, Boneka] --> [Sabun] (confidence: 1.000)
[Sabun, Sprei] --> [Gula] (confidence: 1.000)
[Sabun, Popok] --> [Gula] (confidence: 1.000)
[Boneka] --> [Sabun, Gula] (confidence: 1.000)
[Sabun, Boneka] --> [Gula] (confidence: 1.000)
[Gula, Boneka] --> [Sabun] (confidence: 1.000)
[Sabun, Sprei] --> [Boneka] (confidence: 1.000)
[Boneka] --> [Sabun, Sprei] (confidence: 1.000)
[Sabun, Boneka] --> [Sprei] (confidence: 1.000)
```

CONTOH PENERAPAN PADA PYTHON

klik di sini


TERIMA KASIH