

TUGAS 5 ALGORITMA FP-GROWTH

Mata Kuliah Data Science

Dosen Pengampu : Vynska Amalia Permadi S.Kom., M.Kom



Disusun oleh :

NAMA : Alivi Milova

NIM : 123170062

Kelas : C

PROGRAM STUDI INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

YOGYAKARTA

2020

ALGORITMA FP-GROWTH

A. Pengertian Algoritma FP-Growth

FP-Growth adalah algoritma pencarian *frequent* itemsets yang didapat dari FP-tree dengan menjelajahi *tree* dari bawah menuju ke atas (Tan, Steinbach, & Kumar, 2004). Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki di algoritma FP-Growth. Algoritma ini menentukan *frequent* itemset yang berakhiran suffix tertentu dengan menggunakan metode *divide and conquer* untuk memecah problem menjadi subproblem yang lebih kecil (Han, Kamber, & Pei, Data Mining : Concepts and Techniques, 2011). FP-Growth menggunakan konsep pembangunan *tree* dalam pencapaian *frequent* itemset, tidak menggunakan *generate candidate* seperti Algoritma Apriori. Hal ini lah yang menyebabkan Algoritma FP-Growth lebih cepat dari Algoritma Apriori.

B. Proses Algoritma FP-Growth

FP-Growth merupakan salah satu algoritma yang termasuk dalam association rule mining. (Samuel) Algoritma FP-Growth dibagi menjadi tiga langkah utama, yaitu :

1. Tahap pembangkitan Conditional Pattern Base Conditional Pattern Base merupakan subdatabase yang berisi prefix path (lintasan prefix) dan suffix pattern (pola akhiran). Pembangkitan conditional pattern base didapatkan melalui FP-tree yang telah dibangun sebelumnya.
2. 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 sama dengan minimum support count akan dibangkitkan dengan conditional FP-tree.
3. Tahap pencarian frequent itemset apabila Conditional FP-tree merupakan lintasan tunggal (single path), maka didapatkan frequent itemset dengan melakukan kombinasi item untuk setiap conditional FPtree. Jika bukan lintasan tunggal, maka dilakukan pembangkitan FPgrowth secara rekursif.

C. Contoh Penyelesaian Algoritma FP-Growth

Tujuan dari Algoritma FP-Growth tidak berbeda jauh dengan Algoritma Apriori yaitu untuk mencari Association Rules. Keduanya memiliki perbedaan pada seperti proses Algoritma FP-Growth lebih cepat karena hanya perlu scan database sekali saja sedangkan

proses Algoritma Apriori melakukan scan database berulang kali. Tetapi untuk tingkat akurasi Algoritma Apriori lebih tinggi dibandingkan dengan Algoritma FP-Growth.

Berikut langkah-langkah proses Algoritma FP-Growth :

- a. Data transaksi penjualan diberikan tabel data transaksi sebagai berikut, dengan minimum support count = 2.

:

Id transaksi	Nama item
1	SP Trochess, woods antitusif
2	woods antitusif, vipro G, betadine, decolsin, bodrex
3	SP Trochess, Vipro G, betadine, Hansaplast, kalpanax
4	SP Trochess, betadine, hansaplast
5	Woods antitusif, SP Trochess, minyak tawon, vipro G
6	Vipro G, betadine, SP Trochess, woods antitusif
7	SP Trochess, antangin cair
8	Woods antitusif, vipro G, SP Trochess
9	SP Trochess, woods antitusif, betadine
10	Woods antitusif, vipro G, hansaplast

Tabel 1. Data Transaksi

- b. Frekuensi kemunculan tiap item dapat dilihat pada tabel 2 berikut :

Item	Frekuensi
SP Trochess	8
Woods antitusif	7
Vipro G	6
Betadine	5
Hansaplast	3
Kalpanax	1
Antangin cair	1
Minyak tawon	1
Decolsin	1
Bodrex	1

Tabel 2. Kemunculan tiap item

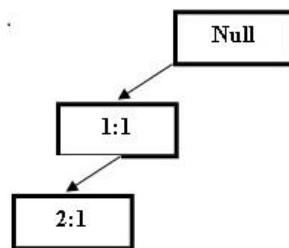
- c. Setelah dilakukan pemindaian pertama didapat item yang memiliki frekuensi diatas support count = 2 . Kelima item inilah yang akan berpengaruh dan akan dimasukkan

kedalam FP-tree, selebihnya dibuang karena tidak berpengaruh signifikan. Berikut tabel kemunculan item diurut berdasarkan yang frekuensinya paling tinggi.

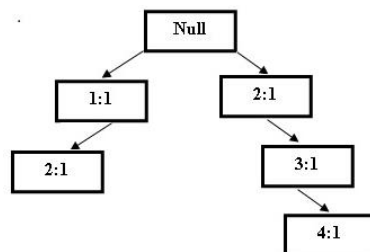
TID	Item
1	{1,2}
2	{2,3,4}
3	{1,3,4,7}
4	{1,4,7}
5	{1,2,3}
6	{1,2,3,4}
7	{1}
8	{1,2,3}
9	{1,2,4}
10	{2,3,7}

Tabel 3. Data Transaksi

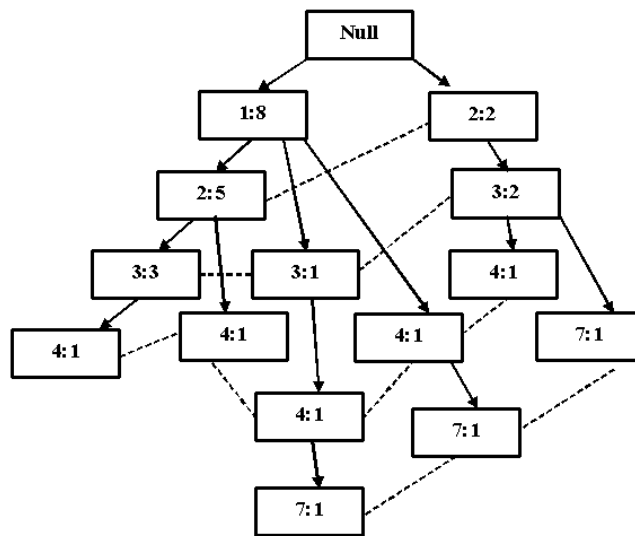
Gambar 2 menunjukkan ilustrasi mengenai pembentukan FP-Tree setelah pembacaan TID 1.



Gambar 2. Hasil pembentukan FP-Tree setelah pembacaan TID 1



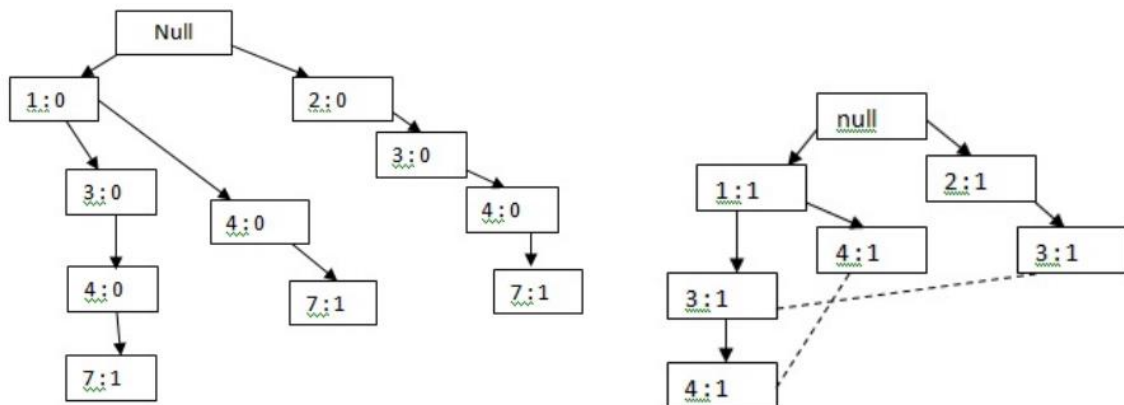
Gambar 3. Hasil Pembentukan FP-Tree setelah pembacaan TID 2



Gambar 4 : Hasil Pembentukan FP-Tree setelah pembacaan TID 10

Gambar 4 menunjukkan proses terbentuknya FP-tree setiap TID dibaca. Setiap simpul pada FP-tree mengandung nama sebuah item dan counter support yang berfungsi untuk menghitung frekuensi kemunculan item tersebut dalam tiap lintasan transaksi. Selanjutnya yaitu mencari semua subset yang memungkinkan dengan membangkitkan conditional FP Tree dan mencari frequent itemset, sesuai urutan frequent list dari yang paling kecil jumlah kemunculannya

a. Kondisi FP-Tree untuk item 7

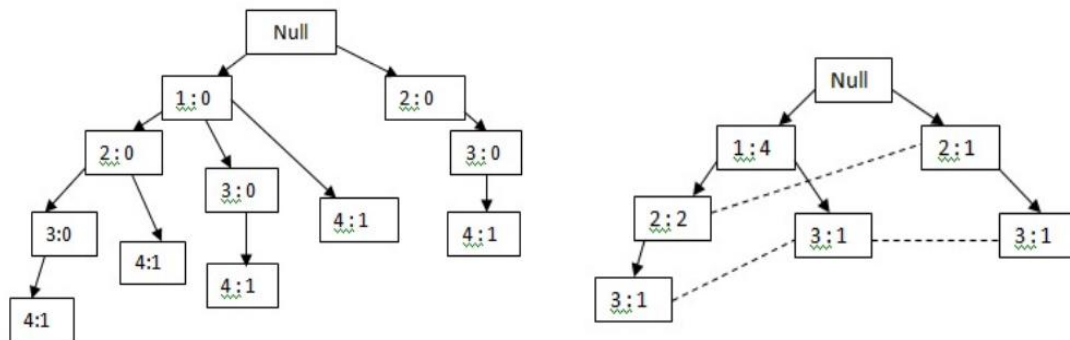


Gambar 5. Kondisi FP-Tree untuk item 7

Ekstrak semua lintasan yang berakhiran 7 dan beri nilai 0 untuk path yang selain 7. Ini untuk mengetahui informasi berapa kali item yang lain dibeli bersamaan dengan item 7 dan mengetahui frequent itemset mana yang memenuhi syarat minimum support. Setelah

itu, naikan satu persatu path 7 sampai ke null dan nilai path 7 dimasukkan ke setiap path yang dilewati sampai ke null. Seperti gambar 7 item 1 dan 2 nilai kemunculan bersamaan dengan item 7 hanya 1 sehingga item 1 dan 2 di buang. Sedangkan item 3 dan 4 nilai kemunculannya 2 maka subset yang dihasilkan adalah $\{7\}$, $\{4,7\}$, $\{3,4,7\}$, $\{3,7\}$.

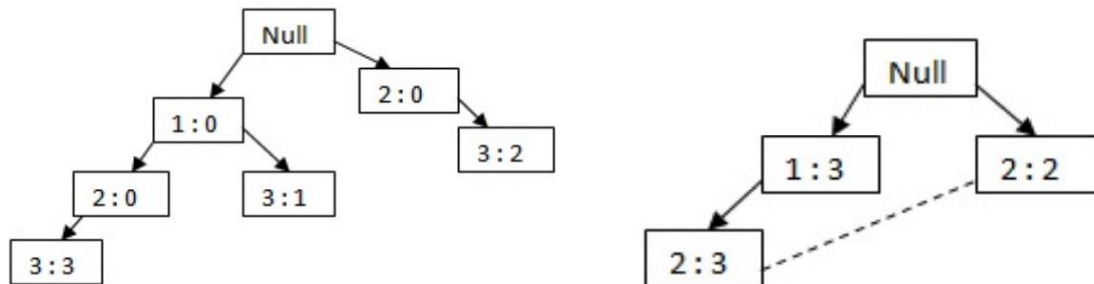
b. Kondisi FP-Tree untuk item 4



Gambar 6. Kondisi FP-Tree untuk item 4

Maka subset yang dihasilkan adalah $\{1,4\}$, $\{1,2,4\}$, $\{1,3,4\}$, $\{2,3,4\}$, $\{2,4\}$, $\{3,4\}$, $\{4\}$.

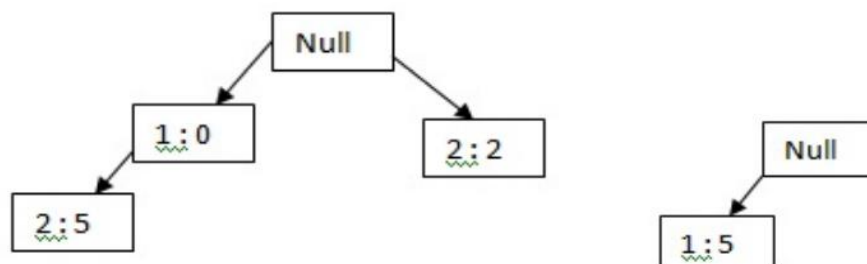
c. Kondisi FP-Tree untuk item 3



Gambar 7. Kondisi FP Tree untuk item 3

Maka subset yang dihasilkan adalah $\{3\}$, $\{2,3\}$, $\{1,2,3\}$, $\{1,3\}$.

d. Kondisi FP-Tree untuk item 2



Gambar 8. Kondisi FP-Tree untuk item 2

Maka subset yang dihasilkan adalah $\{1,2\}$, $\{2\}$.

e. Kondisi FP-Tree untuk item 1



Gambar 9. Kondisi FP-Tree untuk item 1

Pada kasus ini, lintasan yang berakhiran 1 merupakan lintasan tunggal yang berdiri sendiri dan memiliki nilai frekuensi 8 sehingga frequent itemsetnya hanya {1}. Setelah memeriksa kondisi FP Tree didapat 18 frequent itemset yang hasilnya yaitu :

Suffix	Frequent item set
5	{7},{4,7},{3,7},{3,4,7}
4	{4},{3,4},{2,3,4},{1,3,4},{2,4},{1,2,4},{1,4}
3	{3},{2,3},{1,2,3},{1,3}
2	{2},{1,2}
1	{1}

Tabel 4. Hasil frequent item set

Dari 18 frequent itemset tidak semua dihitung karena rule yang dihasilkan adalah jika kita membeli barang A, maka akan membeli barang B, maka subset yang dihitung minimal berisi dua item. Maka yang akan dihitung confidence nya adalah 13 subset, yaitu : {4,7}, {3,7}, {3,4,7}, {3,4}, {2,3,4}, {1,3,4}, {2,4}, {1,2,4}, {1,4}, {2,3}, {1,2,3}, {1,3}, {1,2}. Contoh itemset {2,3,4} yaitu

- $2 \rightarrow 3^4 = 2,3,4 / 2 = 2/7 = 0.28 \times 100 = 28\%$
- $3^4 \rightarrow 2 = 3,4,2 / 3,4 = 2/3 = 66.7 \%$
- $2^4 \rightarrow 3 = 2,4,3 / 2,4 = 2/2 = 100 \%$
- $2 \rightarrow 3 = 2,3 / 2 = 5/7 = 71.4 \%$
- $3 \rightarrow 2 = 3,2 / 3 = 5/6 = 83.3 \%$

Berikut hasil lengkap pola-pola atau rules yang dihasilkan dari 13 subset :

Jika Membeli	Maka akan Membeli	Confidence
1^7 (SP Trochess dan hansaplast)	4 (betadine)	100%
3 (vipro G)	2 (woods antitusif)	83.8 %
4^7 (betadine dan hansaplast)	1 (SP Trochess)	100%
4 (betadine)	1 (SP Trochess)	83.8 %

Tabel 5. Hasil Assosiation Rules