

# SISTEM REKOMENDASI PAKET MAKANAN DENGAN ALGORITMA FP-GROWTH PADA RESTORAN SEAFOOD XYZ

Pahridila Lintang<sup>1)</sup>, Muhammad Iqbal<sup>2)</sup>, Ade Pujianto<sup>3)</sup>

<sup>1), 2, 3)</sup> Teknik Informatika STMIK AMIKOM Yogyakarta

Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281

Email : [pahridila8188@students.amikom.ac.id](mailto:pahridila8188@students.amikom.ac.id)<sup>1)</sup>, [muhammad8238@students.amikom.ac.id](mailto:muhammad8238@students.amikom.ac.id)<sup>2)</sup>,  
[ade.pujianto@students.amikom.ac.id](mailto:ade.pujianto@students.amikom.ac.id)<sup>3)</sup>

## Abstrak

Industri bisnis pada era sekarang cenderung berlandaskan riset agar terpenuhinya kebutuhan pelanggan yang secara otomatis meningkatkan popularitas perusahaan. Pengelola restoran harus cermat dalam mengolah paket yang ada, harus mempunyai kesesuaian dengan perilaku konsumen dalam bertransaksi.

Berdasarkan hal tersebut, kami menawarkan sistem rekomendasi sebagai model aplikasi kami yang dapat melakukan analisis terhadap data transaksi pelanggan. Sistem ini menggunakan algoritma FP-Growth untuk menemukan pola berupa item yang dibeli bersamaan dengan parameter minimum support (nilai penunjang) 0,2 dan confidence (nilai kepastian) yang paling tinggi sebagai bahan untuk rekomendasi paket makanan pada restoran seafood xyz.

**Kata kunci:** Restoran, rekomendasi, FP – Growth.

## 1. Pendahuluan

Saat ini perkembangan restoran di wilayah Yogyakarta semakin meningkat. Bisnis restoran telah mendorong kuat pertumbuhan ekonomi di Yogyakarta, setelah sektor jasa, sektor perdagangan, dan hotel. Oleh karena itu kualitas sistem pelayanan harus selalu ditingkatkan karena merupakan keunggulan kompetitif dalam bisnis. Untuk mengatasi masalah tersebut, Data Mining memberikan solusi dengan menambang informasi dari kumpulan data yang banyak tersimpan untuk menghasilkan pengetahuan yang selama ini jarang diketahui. Pengetahuan tersebut akan membantu dalam mengambil tindakan-tindakan bisnis sebagai upaya pemeliharaan dan peningkatan tingkat kompetitif bisnis restoran (Yusuf, 2006 : E53).

## Perumusan Masalah

1. Bagaimana mengolah data mentah menjadi informasi yang bermanfaat dengan menggunakan teknik data mining.

2. Bagaimana menerapkan algoritma FP-growth pada sistem informasi untuk merekomendasikan kebijakan bisnis pada restoran seafood xyz.

## Batasan Masalah

1. Mengolah data transaksi menjadi rekomendasi paket menggunakan metode FP-Growth
2. Data yang digunakan sebagai contoh hanya 10 data transaksi yang diambil secara urut.
3. Program sistem rekomendasi paket ini digunakan untuk pihak karyawan restoran XYZ.

## Tujuan

Penelitian ini bertujuan untuk :

1. Memanfaatkan data transaksi pada restoran untuk diolah menjadi informasi yang bermanfaat
2. Menerapkan algoritma FP growth dalam sistem Informasi

## Tinjauan Pustaka

Berdasarkan penelitian sebelumnya, telah dirancang sistem rekomendasi yang dibuat oleh Rizky [1], referensi sistem rekomendasi di bidang perpustakaan. Sistem ini menggunakan dialog interaktif juga antara pemakai dengan sistem rekomendasi, yaitu sistem untuk menentukan rekomendasi peminjaman buku pada perpustakaan. daftar transaksi peminjaman buku sebelumnya, kemudian mesin inferensi akan mengolah sehingga akan menghasilkan suatu kesimpulan rekomendasi peminjaman buku. perbedaan dari penelitian sebelumnya adalah pada penelitian sebelumnya masih bersifat *prototype* dan masih di buat berbasis *desktop* sehingga sulit di onlinekan.

## Data Mining

*Data Mining* merupakan suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar [2].

**FP-Growth**

Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FP-Growth [3].

Frequent Pattern Growth (FP-Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent itemset) dalam sebuah kumpulan data [4].

Pada algoritma Apriori diperlukan *generate candidate* untuk mendapatkan *frequent itemsets*.

Akan tetapi, di algoritma FP-Growth *generate candidate* tidak dilakukan karena FP-Growth menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemsets*. Hal tersebutlah yang menyebabkan algoritma FP-Growth lebih cepat dari algoritma Apriori.

Karakteristik algoritma FP-Growth adalah struktur data yang digunakan berbentuk *tree* yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak *frequent itemset* dari FP-Tree.

Penggalian *itemset* yang *frequent* dengan menggunakan algoritma FP-Growth akan dilakukan dengan cara membangkitkan struktur data *tree* atau disebut dengan FP-Tree. Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut [5]:

1. Tahap Pembangkitan *conditional pattern base*,
2. Tahap pebangkitan *conditional FP-Tree*, dan
3. Tahap pencarian *frequent itemset*.

**2. Pembahasan**

a. Pembentukan FP-tree

Untuk menerapkan algoritma FP-growth diawali dengan pembentukan FP-tree. FP-tree memetakan setiap data transaksi ke dalam setiap lintasan tertentu. Karena dalam setiap transaksi yang dipetakan ada kemungkinan transaksi memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa.

Adapun FP-Tree adalah sebuah pohon dengan definisi sebagai berikut :

1. FP-Tree dibentuk dari sebuah akar yang diberi label null.
2. Setiap simpul dalam FP-Tree mengandung tiga informasi penting, yaitu :

- a. Label item : menginformasikan barang.
- b. Support count : merepresentasikan jumlah lintasan transaksi yang melalui simpul tersebut.
- c. Pointer : garis putus – putus yang menghubungkan simpul-simpul dengan label item yang sama antar lintasan.

Contoh diberikan tabel data transaksi yang berisi id untuk mewakili nama item seperti pada tabel 1.

**Tabel 1.**Id setiap item makanan

id	item
1	Nasi
2	Ayam
3	Cumi
4	Kwetiauw
5	Cap cay
6	Kerang
7	Sop
8	Bihun
9	Udang
10	Fuyunghai
11	Mie
12	Seafood

Kemudian tabel 2 berisi sepuluh data transaksi pembelian makanan yang telah melalui proses KDD yang diambil secara urut.

**Tabel 2.**Data Transaksi Pembelian Makanan

No. Transaksi	Item transaksi
1	1, 2, 3,4
2	3, 2, 1
3	6, 4, 9
4	7, 5
5	5, 10
6	5, 8, 9
7	1, 11, 12
8	1, 4, 5
9	1, 13
10	5, 7, 8

Tahap awal adalah melakukan pemindaian terhadap data transaksi yang memiliki nilai *support*  $\geq$  *minimum support* 0,2.

**Tabel 3.**Data Transaksi Dengan Nilai Support  $\geq$  Minimum Support.

Makanan	Frekuensi	Nilai support
1	5	0,5
5	5	0,4

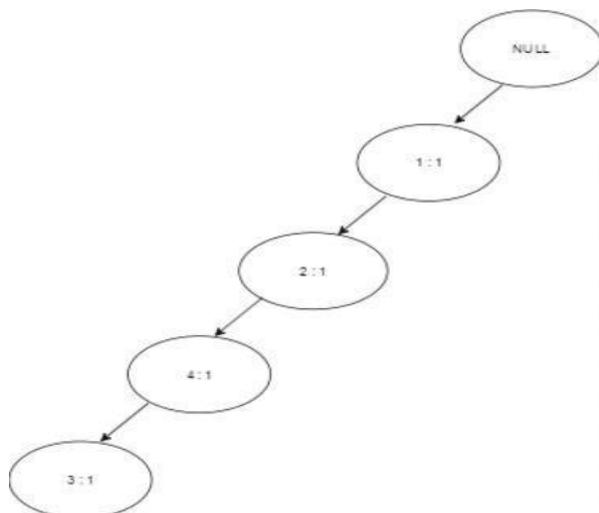
4	3	0,3
2	2	0,2
3	2	0,2
7	2	0,2
8	2	0,2
9	2	0,2

Setelah dilakukan pemindaian pada data transaksi seperti pada tabel 3 selanjutnya adalah menyusun data transaksi berdasarkan frekuensi item untuk pembuatan FP-Tree seperti pada tabel 4.

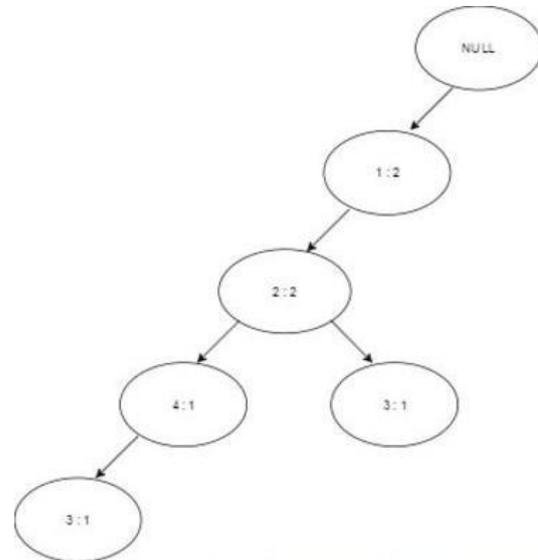
**Tabel 4.** Data Transaksi Yang Diurutkan Sesuai Frekuensi Item

Transaksi	Id makanan
1	1, 2, 4, 3
2	1, 2, 3
3	4, 9
4	5, 7
5	5
6	5, 8, 9,
7	1
8	1, 5, 4
9	1
10	5, 7, 8

Setelah data transaksi diurutkan sesuai *frekuensi item* seperti pada tabel 4, maka gambar 1 akan memberikan ilustrasi mengenai pembentukan FP-tree .

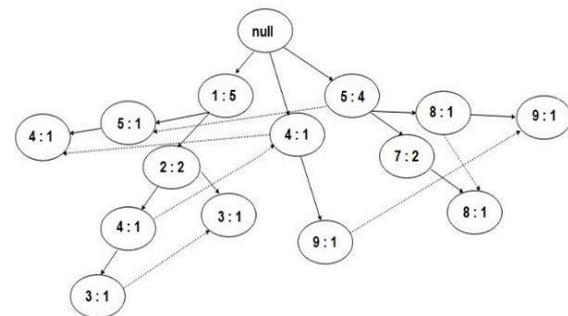


**Gambar 1.** Pembentukan FP-Tree Pada Transaksi pertama



**Gambar 2.** Pembentukan FP-Tree pada transaksi kedua

Seluruh transaksi yang terdapat di tabel 3 dibentuk satu persatu sampai selesai sehingga menjadi FP-Tree dibawah ini



**Gambar 3.** Hasil Pembentukan Fp-Tree Keseluruhan Transaksi 1

**b. Penerapan algoritma FP-growth**

Setelah tahap pembangunan FP-tree selesai, diterapkan algoritma FP-growth untuk mencari *frequent itemset*. 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).
2. Tahap pembangkitan *Conditional FP-tree*

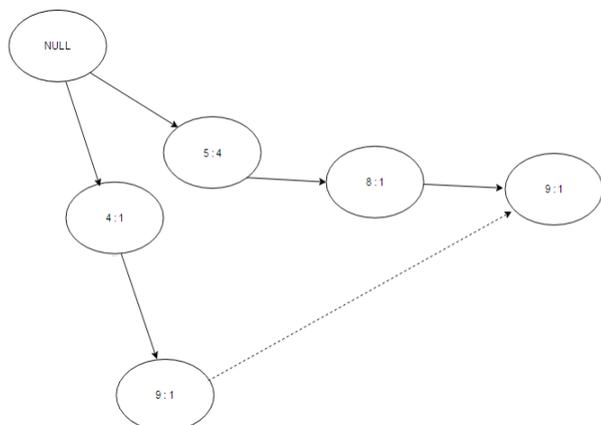
Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah *support count* lebih besar sama dengan minimum *support count* akan dibangkitkan dengan *conditional FP-tree*.

3. Tahap Pencarian *frequent itemset*

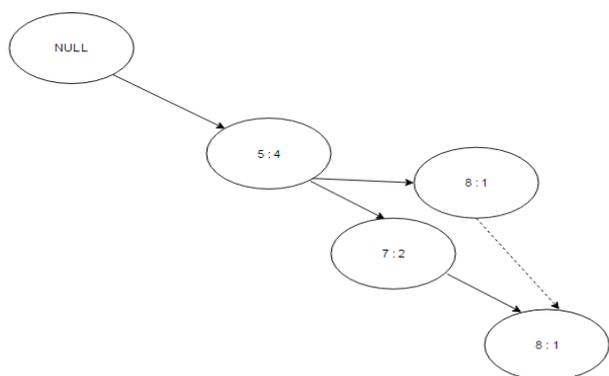
Apabila *Conditional FP-tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi item untuk setiap *conditional FP-tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-growth* secara rekursif.

Akan dicoba menerapkan algoritma *FP-growth* pada data transaksi makanan.

Mengambil *conditional pattern base* pada *FP-Tree* dengan melakukan pemindaian pada *FP-Tree*. Tentukan lintasan dari setiap *item* yang berakhir dengan nilai *support* terkecil, yaitu 9. Berturut-turut dibentuk juga 8,7,3,2,4,5,1. Proses pembentukan dapat dilihat pada gambar berikut.



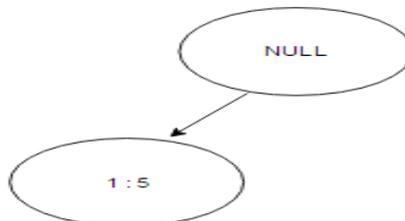
Gambar 4. Lintasan yang Mengandung Simpul 9



Gambar 5. Lintasan yang Mengandung Simpul 8

Seluruh lintasan dari setiap item yang berakhir dengan nilai *support* terkecil, dibentuk satu persatu sampai lintasan yang memiliki nilai *support* 1 dibawah ini :

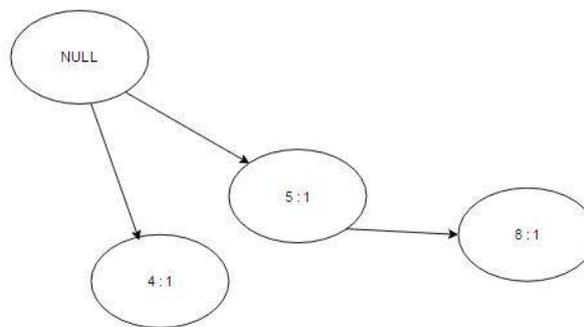
Seluruh lintasan dari setiap item yang berakhir dengan nilai *support* terkecil, dibentuk satu persatu sampai lintasan yang memiliki nilai *support* 1 dibawah ini :



Gambar 6. Lintasan yang Mengandung Simpul 1

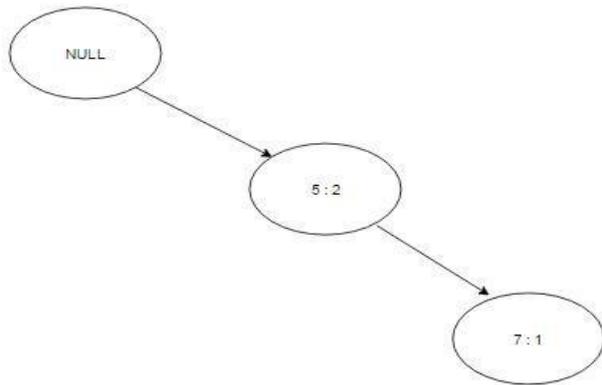
2. Pembangkitan *conditional FP-tree*

Langkah pertama pada simpul 9 di gambar 4 adalah menghapus item 9 dan menghitung ulang *support count*.



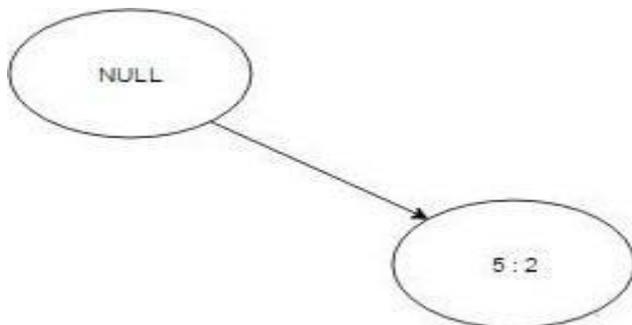
Gambar 7. Setelah item 9 dihilangkan, *support count* simpul di atasnya diperbaharui.

- a. Setelah semua lintasan berakhir di 9, maka simpul 9 dapat dihilangkan. Nilai *support count* di atasnya di perbaharui sesuai dengan kemunculan nilai transaksi dengan item 9 secara bersamaan.
- b. Karena item 9 *frequent*, yaitu nilai *support*  $9 \geq \text{minimum support}$  maka dipecahkan subproblem untuk menemukan *frequent itemset*.
- c. Karena nilai dari *subproblem* 4, 5 dan 8 hanya 1 transaksi maka berdasarkan prinsip *anti-monotone heuristic*, setiap transaksi tidak ada yang *frequent* dan dibuang. Perhitungan di simpul 9 dihentikan dan di lanjutkan pada simpul 8.



**Gambar 8.**Setelah Item 8 Dihilangkan dan Support Count Diatasnya Diperbaharui

- Setelah semua lintasan berakhir di 8, maka simpul 8 dapat dihilangkan. Nilai *support count* di atasnya di perbaharui sesuai dengan kemunculan nilai transaksi dengan item 8 secara bersamaan.
- Karena item 8 *frequent*, yaitu nilai *support*  $9 \geq$  *minimum support* maka dipecah *subproblem* untuk menemukan *frequent itemset*.
- karena nilai dari *subproblem* 7 hanya 1 transaksi maka berdasarkan prinsip *anti-monotone heuristic*, setiap transaksi yang mengandung 7 dibuang. Karena jika *item* 7 tidak *frequent* maka setiap transaksi yang berakhir 7,9 juga tidak *frequent*.



**Gambar 9.**Pohon Prefix yang Berakhir Di 5,9

- Dengan nilai *support* dari 5, didapat bahwa {5,9} termasuk dalam *frequent itemset*.
- Selanjutnya *fp-tree* akan terus melakukan pembangkitan *conditional FP-tree* yang sama sampai ke simpul 1.

Setelah menemukan *frequent itemset* untuk semua akhiran (*suffix*), maka didapat hasil yang dirangkum dalam tabel 5:

**Tabel 5.**Hasil Frequent Itemset

<i>Suffix</i>	<i>Frequent Itemset</i>
8	{8},{5,8}
7	{7},{5,7}
4	{4},{1,4}
3	{3},{1,3},{2,3},{1,2,3}
2	{2},{1,2}

Dengan metode *divide and conquer* ini, maka pada setiap langkah rekursif, algoritma FP-growth akan membangun sebuah *conditional FP-tree* baru yang telah diperbaharui nilai *support count*, dan membuang lintasan yang mengandung item-item yang tidak *frequent* lagi.

### 3. Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari penulisan makalah ini adalah :

- Dengan 10 data transaksi pada tabel 2 dan nilai minimum support 0,2 maka didapat pola pembelian konsumen :
  - Jika membeli ayam maka membeli cumi dengan nilai *confidence* 1. Nilai *confidence* 1 adalah nilai dengan kepastian tertinggi. Rekomendasi paket yang diusulkan adalah : paket 1 menu ayam dan cumi.
  - Jika membeli cap cay maka membeli bihun dengan nilai *confidence* 0,4. Rekomendasi paket yang diusulkan adalah : paket 2 menu cap cay dan bihun.
  - Jika membeli cap cay maka membeli sop dengan nilai *confidence* 0,4. Rekomendasi paket yang diusulkan adalah : paket 3 menu cap cay dan sop.
  - Jika membeli nasi maka membeli kwetiaw dengan nilai *confidence* 0,4. Rekomendasi paket yang diusulkan adalah : paket 4 dengan menu nasi dan kwetiaw.
  - Jika membeli nasi maka membeli cumi dengan nilai *confidence* 0,4. Rekomendasi paket yang diusulkan adalah : paket 5 dengan menu nasi dan cumi.
  - Jika membeli nasi maka membeli ayam dan cumi dengan nilai *confidence* 0,4. Rekomendasi paket yang diusulkan adalah : paket 6 dengan menu nasi, ayam dan cumi.
- Algoritma fp – growth menutupi kelemahan algoritma apriori dalam menentukan frequent itemset. Tahap menentukan frequent itemset sangat berpengaruh terhadap performa data mining, terutama jika kandidat itemset berjumlah sangat besar. Pada algoritma fp-growth hanya di lakukan dua kali scan database yaitu

pada awal iterasi untuk menentukan support count dan pada pembacaan TID pertama. Dengan begitu performa data mining akan meningkat dibanding algoritma apriori yang akan melakukan scan database sebanyak k-itemset.

### **Daftar Pustaka**

- [1] Mei, Rizki. 2010. *Perbandingan Algoritma Apriori dan Algoritma Fp-Growth Untuk Perekomendasi Pada Transaksi Peminjaman Buku di Perpustakaan Universitas Dian Nuswantoro*. Skripsi, Prodi Teknik Informatika : Universitas Dian Nuswantoro Semarang.
- [2] Kusri dan E. T. Lutfi. 2009. *Artificial Intelligent (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu. p.109
- [3] Asiyah, S. (2005). *Sistem Pakar Diagnosa Penyakit Gigi*. Skripsi, Fakultas MIPA :Universitas Gadjah Mada.
- [4] Y. Wibisono. *Metode Statistik. 1*. Yogyakarta : Andi. 2009 : 45.
- [5] WHO. Health Topics Lung. 2014.

### **Biodata Penulis**

**Pahridila Lintang**, saat ini menjadi mahasiswa di STMIK AMIKOM Yogyakarta.

**Muhammad Iqbal**, saat ini menjadi mahasiswa di STMIK AMIKOM Yogyakarta.

**Ade Pujianto**, saat ini menjadi mahasiswa di STMIK AMIKOM Yogyakarta