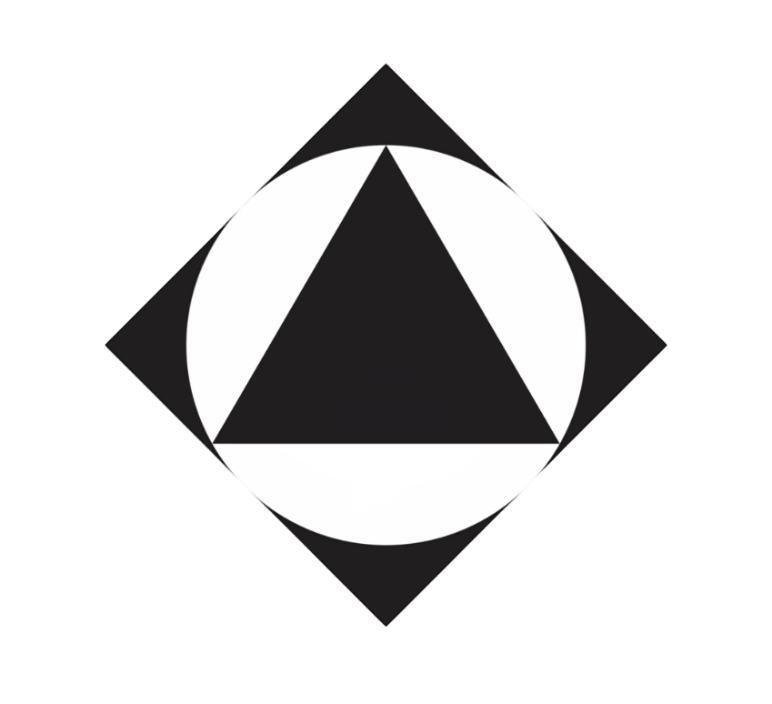
**Penerapan Algoritma FP-Growth Dalam Analisis Pola Asosiasi pada Penjualan Supermarket**



**Disusun Oleh:**

Emelsha Viadra 152022056

Naufal Zaidan 152022168

Prodi Informatika

Fakultas Teknologi Industri

Institut Teknologi Nasional

Bandung

2024

FP-Growth (Frequent Pattern Growth) adalah algoritma yang digunakan dalam data mining untuk menemukan pola frekuen dalam basis data yang besar dan kompleks. Algoritma ini dirancang untuk mengidentifikasi item-item yang sering muncul bersamaan dalam transaksi, sehingga dapat digunakan untuk analisis pola asosiasi, seperti market basket analysis.

Berbeda dengan algoritma tradisional seperti Apriori, yang memerlukan banyak pemindaian data dan menghasilkan kandidat pola yang besar, FP-Growth menggunakan pendekatan yang lebih efisien. Algoritma ini pertama-tama membangun sebuah struktur data yang disebut FP-Tree (Frequent Pattern Tree), yang merepresentasikan item-item frekuen dalam bentuk pohon. Dengan menggunakan FP-Tree, FP-Growth dapat mengeksplorasi semua pola frekuen tanpa harus menghasilkan semua kandidat pola terlebih dahulu, sehingga mengurangi waktu dan sumber daya yang diperlukan untuk pemrosesan.

Kelebihan utama dari FP-Growth adalah kemampuannya untuk menangani dataset yang besar dengan lebih efisien, membuatnya ideal untuk digunakan dalam analisis data transaksi di berbagai domain, termasuk retail dan e-commerce. Melalui penerapan FP-Growth, perusahaan dapat mengungkapkan pola-pola pembelian pelanggan yang berharga, yang pada gilirannya dapat digunakan untuk meningkatkan strategi pemasaran, pengelolaan persediaan, dan pengalaman pelanggan.

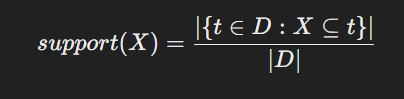
**Garis Tahun Sejarah FP-Growth**

* **2000**:
  + **Perkenalan FP-Growth**: FP-Growth diperkenalkan oleh Jiawei Han, Jian Pei, dan Yiwen Yin dalam makalah berjudul "Mining Frequent Patterns without Candidate Generation." Algoritma ini dirancang untuk mengatasi kelemahan algoritma Apriori dalam menemukan pola frekuen dengan lebih efisien.
* **2001**:
  + **Adopsi Awal**: Algoritma FP-Growth mulai diadopsi dalam aplikasi nyata di industri, terutama dalam analisis data transaksi di sektor retail dan e-commerce.
* **2002-2005**:
  + **Pengembangan dan Variasi**: Penelitian lebih lanjut menghasilkan variasi dari FP-Growth yang meningkatkan efisiensinya, seperti adaptasi untuk data streaming dan konteks lain yang memerlukan kecepatan pemrosesan.
* **2006**:
  + **Integrasi dengan Alat Analisis Data**: FP-Growth mulai diintegrasikan ke dalam berbagai alat dan platform analisis data, memudahkan pengguna untuk menerapkan algoritma ini dalam proyek data mining mereka.
* **2010**:
  + **Penggunaan dalam Big Data**: Dengan pertumbuhan data besar, FP-Growth tetap relevan dan digunakan dalam aplikasi big data, didukung oleh kemajuan dalam teknologi penyimpanan dan pemrosesan data.
* **2015-sekarang**:
  + **Penelitian Berkelanjutan**: FP-Growth terus menjadi fokus penelitian dalam komunitas data mining, dengan inovasi baru yang menjelajahi efisiensi algoritma dan penerapannya di berbagai domain, termasuk kesehatan, analisis web, dan industri lainnya.

**Rumus dan Konsep dalam FP-Growth**

Algoritma FP-Growth berfokus pada penemuan pola frekuen dalam dataset transaksi. Berikut adalah rumus dan konsep kunci yang digunakan dalam algoritma ini:

1. **Support dan Frequent Itemset**

* **Support**: Support mengukur seberapa sering suatu itemset muncul dalam database transaksi. Rumus untuk menghitung support suatu itemset X adalah:

Di mana:

* + |D| adalah jumlah total transaksi dalam database.
  + {t ∈ D : X ⊆ t} adalah himpunan transaksi yang mengandung itemset X.
* **Frequent Itemset**: Sebuah itemset dianggap **frequent** jika support-nya melebihi ambang batas yang ditentukan, yaitu min\_support:



1. **Pembangunan FP-Tree**

FP-Growth membangun sebuah struktur data yang disebut **FP-Tree** untuk menyimpan item frekuen dan relasinya dalam bentuk pohon. Proses pembangunannya meliputi langkah-langkah berikut:

1. **Hitung frekuensi setiap item**: Hitung jumlah kemunculan setiap item dalam database.
2. **Buang item yang tidak memenuhi batas**: Hapus item yang tidak memenuhi batas minimum support.
3. **Urutkan item**: Urutkan item berdasarkan frekuensi kemunculannya dari yang tertinggi hingga terendah.
4. **Bangun FP-Tree**: Tambahkan setiap transaksi ke dalam FP-Tree sesuai dengan urutan item yang telah ditentukan.

**3. Ekstraksi Pola Frekuen**

Setelah FP-Tree dibangun, langkah selanjutnya adalah mengekstrak pola frekuen dengan cara berikut:

1. **Mining dengan cara rekursif**: Untuk setiap item a dalam FP-Tree, lakukan mining pada subtree yang terhubung dengan a.
2. **Identifikasi pola frekuen**: Pola frekuen yang dihasilkan akan mencakup semua item yang sering muncul bersama dengan item a.

**Studi Kasus**

|  |  |
| --- | --- |
| **DATASET** | |
| **TID** | **Item** |
| 100 | Cola, Kacang, Telur |
| 200 | Kacang, Popok |
| 300 | Kacang, Kopi |
| 400 | Cola, Kacang, Popok |
| 500 | Cola, Kopi |
| 600 | Kacang, Kopi |
| 700 | Cola, Kopi |
| 800 | Cola, Kacang, Kopi, Telur |
| 900 | Cola, Kacang, Kopi |

* Menentukan Minimum support, kami menggunakan minimum support 2
* Mencari frequent items

|  |  |
| --- | --- |
| **Frequent Itemset** | |
| **Item** | **Total** |
| Kacang | 7 |
| Cola | 6 |
| Kopi | 6 |
| Popok | 2 |
| Telur | 2 |

* Mengurutkan Priority

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Urutan Priority** | | | |
| 100 | Kacang | Cola | Telur |  |
| 200 | Kacang | Popok |  |  |
| 300 | Kacang | Kopi |  |  |
| 400 | Kacang | Cola | Popok |  |
| 500 | Cola | Kopi |  |  |
| 600 | Kacang | Kopi |  |  |
| 700 | Cola | Kopi |  |  |
| 800 | Kacang | Cola | Kopi | Telur |
| 900 | Kacang | Cola | Kopi |  |

* Membuat Conditional FP Tree

|  |  |  |  |
| --- | --- | --- | --- |
| **Conditional FP Tree** | | | |
| Telur | Kacang: 2 | Cola: 2 | Kopi: 1 |
| Popok | Kacang: 2 | Cola: 1 |  |
| Kopi | Kacang: 4 | Cola: 4 |  |
| Cola | Kacang: 4 |  |  |

* Frequent Pattern

|  |  |  |
| --- | --- | --- |
| **Frequent Pattern** | | |
| Telur | {Kacang, Telur} | 2 |
| {Cola, Telur} | 2 |
| Popok | {Kacang, Popok} | 2 |
| Kopi | {Kacang, Kopi} | 4 |
| Cola | {Cola, Kopi} | 4 |
| {Kacang, Cola} | 4 |

* **FP Tree**