

# Aplikasi BFS pada *Frequent Subgraph Mining* dalam Analisis Database Transaksi

Erma Safira Nurmasiyita (13516072)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13516072@std.stei.itb.ac.id

**Abstract**—Representasi graf telah banyak digunakan untuk memodelkan berbagai struktur informasi yang kompleks, seperti sirkuit, struktur molecular, web network, basisdata multirelasional, dan sebagainya. Seiring berjalannya waktu, struktur ini dapat berkembang menjadi struktur yang lebih rumit. Sebagai solusinya, dikembangkan pula metode graf mining. Tujuan utama dari graf mining adalah untuk memperoleh pola-pola menarik pada data yang menghasilkan informasi baru. Salah satu operasi dari graf mining adalah *frequent subgraph mining*. Subbidang ini merupakan kajian yang vital pada *graph mining* untuk mendeteksi pola-pola yang sering muncul pada data graf. Paper ini akan menjelaskan dan menganalisis mengenai *frequent subgraph mining* yang memanfaatkan algoritma Apriori dengan konsep *Breadth-First Search*. Kemudian juga dibahas mengenai pemanfaatan *frequent subgraph mining* dalam analisis bisnis untuk memprediksi produk yang diminati pelanggan, sehingga dapat menyusun strategi penjualan yang baik.

**Keywords**—*graph mining; Apriori; frequent subgraph; Breadth-First Search; support*

## I. PENDAHULUAN

Sebagai salah satu struktur data dasar, graf merupakan struktur data yang umum digunakan dalam memodelkan interaksi yang rumit dengan aplikasi yang luas, seperti kimia molekular, bioinformatika, *computer vision*, indexing, retrieval, dan analisis web. Banyak struktur graf dalam penelitian sains dan aplikasi komersial yang memiliki pola yang rumit.

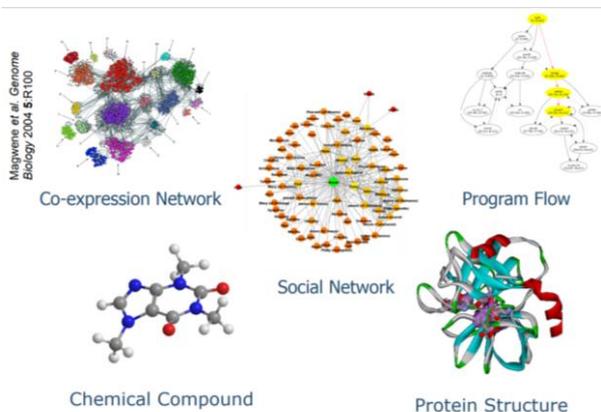


Fig 1: Berbagai graf yang digunakan untuk memodelkan persoalan

(Sumber: [https://www.ethz.ch/content/dam/ethz/special-interest/bse/borgwardt-lab/documents/slides/CA10\\_GraphMining.pdf](https://www.ethz.ch/content/dam/ethz/special-interest/bse/borgwardt-lab/documents/slides/CA10_GraphMining.pdf))

Graf yang telah menghubungkan banyak simpul dapat membentuk koneksi ke jaringan lain yang berbeda, seperti jaringan telekomunikasi, jaringan computer, dan jaringan biologi. Bentuk graf yang kian makin rumit merupakan tantangan tersendiri bagi para pengembang.

Untuk mengatasi kompleksitas analisis graf, dikembangkanlah metode graf mining. Graf mining merupakan langkah identifikasi graf yang berfokus pada ekstraksi informasi yang dapat diperoleh dari graf kompleks.

Tujuan utama dari graf mining adalah untuk menemukan informasi baru berupa pola-pola menarik yang dapat muncul pada dataset graf. Pencarian pola pada graf untuk mengkarakterisasikan, mendiskremerinasikan, dan mengklasifikasi merupakan hal yang vital dalam mendefinisikan data. Informasi ini berguna untuk mengambil tindakan selanjutnya yang dapat dilakukan untuk mengoptimalkan sistem yang dianalisis.

## II. DASAR TEORI

### A. Teori Graf

Graf adalah struktur diskrit yang mengandung sekumpulan objek yang dihubungkan dengan busur. Graf digunakan untuk merepresentasikan objek diskrit beserta hubungan antara objek tersebut.

Secara matematis, graf didefinisikan sebagai pasangan himpunan  $(V,E)$  dituliskan dengan notasi  $G = (V,E)$ , yang dalam hal ini:

$V$  adalah himpunan tidak kosong dari simpul (vertices atau nodes)  $= \{v_1, v_2, \dots, v_n\}$ ,

$E$  adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul  $= \{e_1, e_2, \dots, e_n\}$

(Rinaldi Munir, 2006 : VIII – 2).

$V$  dinyatakan tidak boleh kosong, sedangkan  $E$  boleh kosong, sehingga sebuah graf dimungkinkan tidak memiliki sisi, tetapi simpulnya harus ada minimal satu. Sebuah graf yang hanya terdiri dari sebuah simpul dinamakan Graf Trivial.

Secara geometris, graf digambarkan sebagai kumpulan titik (simpul) di dalam sebuah bidang yang dihubungkan dengan sekumpulan garis (sisi).

Pasangan simpul dapat dihubungkan oleh dua atau lebih sisi, yang disebut sisi-ganda (multiple edges atau parallel edges). Sedangkan sisi yang berawal dan berakhir pada simpul yang sama yang membentuk sebuah lingkaran disebut gelang atau kalang (loop).

### 2.1. Terminologi Graf

Berikut merupakan terminologi atau istilah yang dipakai pada teori graf:

- a. Bertetangga (Adjacent), sepasang simpul dikatakan bertetangga apabila kedua simpul terhubung langsung dengan sebuah sisi yang sama.
- b. Bersisian (Incident), sebuah sisi dikatakan bersisian dengan dua buah simpul yang dihubungkan olehnya. Untuk sembarang sisi  $e = (v_1, v_2)$ , sisi  $e$  bersisian dengan simpul  $v_1$  dan  $v_2$ .
- c. Simpul Terpencil (Isolated Vertex), sebuah simpul dikatakan terpencil apabila simpul tersebut tidak mempunyai tetangga, yang berarti tidak bersisian dengan sisi manapun.
- d. Graf Kosong (Null Graph), graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong, dengan jumlah simpul sebanyak  $n$ .
- e. Derajat (Degree), derajat sebuah simpul pada graf tidak berarah merupakan jumlah sisi yang bersisian dengan simpul tersebut. Sedangkan pada graf berarah, derajat simpul  $v$  dibedakan menjadi menjadi dua sesuai representasi arahnya, yaitu:
  - 1) Derajat Masuk ( $d_{in}(v)$ ): jumlah busur yang masuk ke dalam simpul  $v$ .
  - 2) Derajat Keluar ( $d_{out}(v)$ ): jumlah busur yang keluar dari simpul  $v$ .
  - 3) Derajat total simpul ( $d(v)$ ): jumlah busur masuk dan keluar.
- f. Lintasan (Path), lintasan adalah barisan selang-seling simpul dengan sisi dimulai dari simpul awal  $v_0$  ke simpul tujuan  $v_n$ . Panjang lintasan adalah jumlah sisi dari simpul awal ke simpul tujuan dalam graf  $G$ .
- g. Siklus (Cycle) atau Sirkuit (Circuit), siklus adalah lintasan yang berawal dan berakhir pada simpul yang sama. Panjang sirkuit adalah jumlah sisi pada sirkuit.
- h. Terhubung (Connected), graf disebut terhubung apabila setiap pasang  $v_i$  dan  $v_j$  pada graf mempunyai lintasan yang menuju satu sama lain (terdapat lintasan dari  $v_i$  ke  $v_j$  dan sebaliknya). Jika tidak, maka graf disebut graf tak terhubung. Pada graf berarah, terdapat istilah terhubung kuat dan terhubung lemah. Terhubung kuat berarti apabila terdapat lintasan berarah dari  $u$  ke  $v$ , maka terdapat lintasan berarah dari  $v$  ke  $u$  pula. Sedangkan terhubung lemah berarti

apabila terdapat busur dari  $u$  ke  $v$  tetapi tidak terdapat busur  $v$  ke  $u$ .

- i. Upagraf (Subgraph) dan Komplemen Upagraf, upagraf adalah graf yang seluruh anggota simpul dan sisinya merupakan anggota dari himpunan simpul dan sisi graf lainnya. Komplemen dari sebuah upagraf adalah graf yang anggota himpunan simpul dan sisinya terdiri dari sisi dan simpul yang ada dalam graf  $G$  namun tidak dalam upagraf tersebut.
- j. Upagraf Merentang (Spanning Subgraph). Sebuah graf merupakan upagraf merentang graf lainnya ketika elemen himpunan simpulnya terdiri dari seluruh elemen himpunan simpul graf lainnya, dengan himpunan sisi yang berbeda.
- k. Cut-set, merupakan himpunan sisi dari graf terhubung yang apabila dibuang akan menyebabkan graf tersebut tidak lagi terhubung.
- l. Graf Berbobot, adalah graf yang tiap sisinya memiliki nilai.

### B. Algoritma Breadth-First Search

Algoritma *Breadth-First Search* merupakan algoritma penelusuran graf yang mengunjungi simpul secara sistematis dengan melakukan pencarian secara melebar. Strategi ini mendahulukan kunjungan ke simpul-simpul yang bertetangga, kemudian proses diulangi untuk simpul-simpul tetangganya.

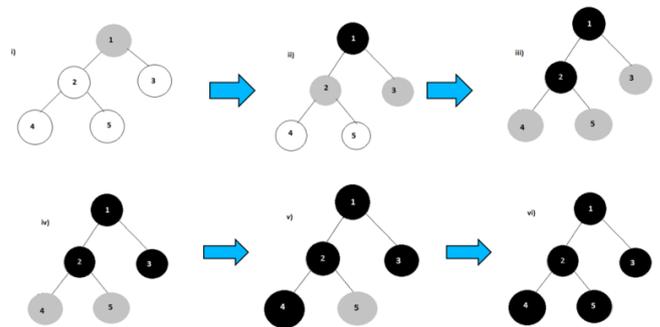


Fig 2: Penelusuran graf secara BFS  
(Sumber: <https://hadooptutorial.wikispaces.com/Iterative+MapReduce+and+Counters>)

Dalam implementasinya, algoritma BFS menggunakan tipe data *queue* untuk menyimpan simpul-simpul yang akan dikunjungi berikutnya. Penelusuran dimulai dari simpul start yang ditentukan. Berikut merupakan langkah-langkah penelusuran dengan BFS:

- Langkah 1: Kunjungi simpul tetangga yang belum dikunjungi. Tandai simpul yang telah dikunjungi dan masukkan ke antrian.
- Langkah 2: Jika tidak terdapat simpul yang bertetangga, hapus simpul dari antrian.
- Langkah 3: Ulangi Langkah 1 dan 2 hingga antrian kosong.

### C. Graph Mining

Graph mining merupakan proses mengekstraksi pengetahuan dari dataset yang dimodelkan dalam *graph*. Objektif utama Graph Mining adalah menemukan keterhubungan implisit antar data pada suatu dataset dengan menganalisis link-link yang direpresentasikan dalam graf sebagai busur (*edge*). Kumpulan data yang memiliki hubungan diklasifikasikan menjadi beberapa kelompok subgraph. Kelompok-kelompok subgraph ini dapat membentuk pola-pola yang dapat dianalisis lebih lanjut dengan *pattern mining*.

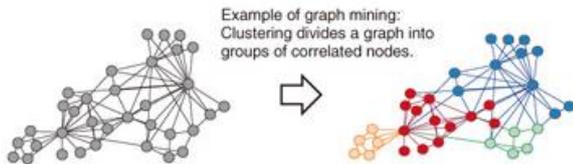


Fig 3: Contoh proses graf mining dalam mengklasifikasi data yang berkorelasi. Data direpresentasikan dengan simpul. (Sumber: [https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201312ra1\\_s.html](https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201312ra1_s.html))

Contoh operasi graph mining adalah *Frequent Subgraph Mining*, *Community Detection*, dan *Clustering*. Masing-masing operasi ini bergantung pada aspek yang berbeda dari input graf, dan properti graf yang berbeda mempengaruhi kebenaran dan performa dari operasi.

Graf yang digunakan pada analisis berupa subgraph sampling. Sampling adalah teknik untuk menghasilkan grafik perwakilan yang lebih kecil yang dapat digunakan untuk tujuan analisis. Jumlah dan metode sampling yang dipilih harus disesuaikan dengan graf pada populasi untuk menghindari biasanya informasi yang diperoleh.

### D. Pattern Mining

*Pattern mining* merupakan salah satu operasi dari *graph mining* yang berfokus pada penelusuran *data mining* untuk mencari pola menarik, informasi tak terduga, dan berguna pada dataset. Definisi pola menarik dapat memiliki banyak arti, seperti pola yang sering muncul dalam set data, pola dengan *confidence* tinggi, maupun pola yang paling langka.

*Pattern mining* dapat dimodelkan dengan berbagai struktur data seperti basisdata transaksi, database *sequence*, data spasial, hingga graf. Algoritma pencariannya dapat diaplikasikan dengan berbagai jenis pola, seperti subgraph, asosiasi, *indirect associations*, pola periodic, pola sekuensial, dan lainnya.

Meskipun subgraph isomorfis merupakan salah satu permasalahan NP-complete, terdapat sejumlah tools untuk mencari pola *frequent subgraph* (contohnya gSpan, GASTON). Kakas ini efektif untuk diterapkan pada database yang besar, dengan pertimbangan desain database itu sendiri.

## III. METODE FREQUENT SUBGRAPH MINING

*Frequent subgraph mining* merupakan proses untuk mengidentifikasi pengulangan substruktur pada database. Database dapat berupa koleksi sejumlah graf kecil, maupun sebuah graf berukuran besar. Metode ini digunakan untuk mengurai dan mengklasifikasi set graf, membedakan graf dari kelompok graf lainnya, membangun indeks graf, hingga memfasilitasi pencarian pada graf set data.

Tujuan dari proses ini adalah untuk menemukan pola berupa subgraph yang berulang pada graf dataset paling sedikit  $k$  kali. Parameter  $k$  ini disebut sebagai *minimum support*. Sebuah subgraph disimpulkan *frequent* apabila subgraph ini muncul paling sedikit  $k$  kali.

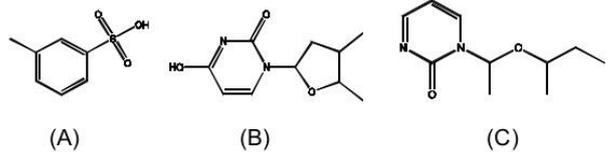
Pada database yang berupa koleksi graf kecil, jumlah kemunculan atau yang disebut sebagai *support* adalah  $n$  apabila subgraph ini muncul sebanyak  $n$  pada database.

Penyelidikan pola *subgraph* yang sering muncul terdiri dari dua langkah. Pada langkah pertama, dibentuk kandidat *frequent subgraph*. Kemudian pada langkah kedua, mengecek frekuensi kemunculan setiap kandidat. Mayoritas studi pada *frequent subgraph mining* berfokus pada optimalisasi langkah pertama, karena metode langkah kedua mengandung tes isomorfis subgraph yang memiliki kompleksitas tinggi (permasalahan NP-complete).

Berikut merupakan contoh penerapan frequent pattern mining beserta hasilnya dalam mendeteksi molekul senyawa kimia. Ditentukan nilai minimal supportnya adalah 2.

### Example: Frequent Subgraphs

#### GRAPH DATASET



#### FREQUENT PATTERNS (MIN SUPPORT IS 2)



Fig 4: Contoh frequent pattern pada bidang kimia organik dengan nilai  $\text{min\_sup} = 2$

(Sumber: <https://www.slideshare.net/HouwLiong/lect12-graph-mining>)

Metode penyelidikan pola *subgraph* memiliki banyak variasi sesuai strategi pendekatannya. Salah satunya adalah pendekatan berbasis *Apriori* yang memanfaatkan konsep *Breadth-First Search*. Berikut merupakan analisis subgraph berbasis *Apriori*.

Algoritma *frequent substructure mining* memiliki karakteristik yang sama dengan algoritma *frequent mining* pada *item set*. Pencarian subgraf yang sering muncul dimulai dengan graf berukuran kecil dan memproses secara *bottom-up* dengan membentuk simpul, busur, maupun lintasan pada kandidat.

*Frequent substructure*  $G$  adalah subgraph yang koheren apabila informasi antara  $G$  dan tiap subgraphnya diatas acuan atau *threshold*. Jumlah substruktur yang koheren lebih sedikit daripada *frequent substructure*. Maka dari itu, graf mining pada substruktur koheren dapat memangkas *redundancy* pada pola (contoh: pola yang memiliki kesamaan satu sama lain dan memiliki nilai *support*). Sebuah metode yang mangkus telah dikembangkan untuk substruktur ini,

#### A. Algoritma Apriori

Algoritma Apriori menggunakan strategi *pattern mining* klasik dengan konsep *Breadth-First Search*. AprioriGraph menggunakan metodologi *level-wise mining*, yaitu melakukan analisis dengan sejumlah tahapan iteratif. Dalam prosesnya, algoritma ini membutuhkan struktur hash tree untuk menghitung jumlah kemunculan kandidat itemset.

Pada setiap iterasi, ukuran dari *frequent substructure* yang baru ditambahkan dengan satu. Substruktur baru ini dibentuk dengan menggabungkan dua substruktur yang diperoleh dari iterasi sebelumnya. Jumlah kemunculan dari substruktur ini dihitung. Jumlah kemunculan yang besar digunakan kembali untuk membuat kandidat substruktur yang lebih besar pada iterasi berikutnya.

Berikut merupakan *framework* umum dari Algoritma AprioriGraph.  $S_k$  merupakan *frequent sub-structure* dengan ukuran  $k$ .

*Frequent subgraph mining* dengan Algoritma Apriori.

*Input:*

- $D$ , a graph data set;
- $Min\_sup$ , the minimum support threshold.

*Output:*

- $S_k$ , the frequent substructure set.

*Method:*

$S_1 \leftarrow$  frequent single-elements in the data set;  
Call AprioriGraph( $D$ ,  $min\_sup$ ,  $S_1$ );

procedure AprioriGraph( $D, min\_sup, S_k$ )

- (1)  $S_{k+1} \leftarrow \emptyset$ ;
- (2) for each frequent  $g_i \in S_k$  do
- (3)   for each frequent  $g_j \in S_k$  do
- (4)     for each size  $(k+1)$  graph  $g$  formed by the merge of  $g_i$  and  $g_j$  do
- (5)       if  $g$  is frequent in  $D$  and  $g \in S_{k+1}$  then
- (6)         insert  $g$  into  $S_{k+1}$ ;
- (7) if  $S_{k+1} \neq \emptyset$  then
- (8)   AprioriGraph( $D$ ,  $min\_sup$ ,  $S_{k+1}$ );

(9) return;

Dalam proses pemindaian, untuk setiap item pada dataset dihitung variable *support(sup)* yang diperoleh dari jumlah kemunculan setiap item pada seluruh record. Sebagai acuan/threshold, ditentukan nilai minimal *support* ( $min\_sup$ ) terlebih dahulu pada awal eksekusi.

Pada strategi Apriori, dilakukan beberapa tahap iterasi dalam dataset. Tiap iterasi dilakukan pemindaian untuk mendapatkan nilai *support* dari setiap item dan menghasilkan *frequent pattern*. Nilai *support* adalah jumlah kemunculan item pada setiap *record* data. Setelah *support* dari setiap item didapat, item yang memiliki *support* diatas *minimum support* ( $min\_sup$ ) dipilih sebagai pola *frequent* dengan panjang satu (1-itemset).  $K$ -itemset adalah istilah untuk satu set yang terdiri dari  $k$  item. Kemudian pada iterasi kedua akan menghasilkan 2-itemset, dimana tiap setnya terdiri dari dua item.

1. Pada iterasi pertama (level  $k=1$ ), yaitu dilakukan pemindaian pada dataset untuk memperoleh *frequent 1-itemset*.
2. Untuk setiap level  $k$ :
  - Bentuk kandidat pola dengan panjang  $(k+1)$  dari *frequent pattern* level  $k$
  - Pindai dataset dan pangkas kandidat yang tidak memenuhi minimum *support*.

#### B. Prinsip Algoritma Apriori

Pada algoritma Apriori, subset dari *frequent pattern* juga merupakan *frequent pattern*. Contohnya jika terdapat *frequent pattern*  $\{A, B, C\}$ , maka  $\{A,B\}$ ,  $\{A\}$ ,  $\{B,C\}$ ,  $\{B\}$ ,  $\{A,C\}$ ,  $\{C\}$  juga termasuk *frequent pattern*. Jika ada pola  $X$  yang tidak sering muncul (*infrequent*), maka superset dari pola  $X$  tidak perlu dibentuk.

### IV. PENERAPAN DALAM ANALISIS TRANSAKSI

*Frequent pattern mining* dapat digunakan untuk menganalisis database transaksi pembelian. Metode ini bekerja dengan melihat pola item apa saja yang cenderung dibeli secara bersamaan oleh pelanggan. Pengusaha dapat melakukan strategi bisnis dengan memanfaatkan informasi ini, sehingga dapat meningkatkan daya beli pelanggan.

Sebagai contoh penerapannya, misalkan terdapat data transaksi sederhana minimarket sebagai berikut. Ditentukan nilai minimum *support*nya adalah 2.

Transaksi	Item
T1	Garam, kremer, teh
T2	Kopi, kremer, gula
T3	Garam, kopi, kremer, gula
T4	Kopi, gula

Tabel 1: Tabel data pembelian item pada setiap transaksi

Untuk setiap kemunculan item, dihitung nilai kemunculannya(support). Berikut merupakan hasil dari iterasi pertama dengan nilai itemset 1:

Item	Support
Garam	2
Kopi	3
Kremer	3
Teh	1
Gula	3

Tabel 2: Hasil iterasi 1 dengan nilai itemset 1

Item yang memiliki nilai support dibawah minimum support tidak perlu diikuti kembali pada iterasi berikutnya. Item yang dipangkas di-highlight dengan warna abu-abu.

Berikut merupakan hasil dari iterasi ke-dua dengan mengkombinasikan item yang masih tersedia untuk itemset 2:

Item	Support
{garam, kopi}	1
{garam, kremer}	2
{garam, gula}	1
{kopi, kremer}	2
{kopi, gula}	3
{kremer, gula}	2

Tabel 2: Hasil iterasi 2 dengan nilai itemset 2

Berikut merupakan hasil iterasi ke-3 dengan kombinasi dari pattern sebelumnya untuk itemset 3:

Item	Support
{kopi, kremer, gula}	2

Tabel 3: Hasil iterasi 3 dengan nilai itemset 3

Dalam implementasinya, jenis item pada database disimpan sebagai simpul. Simpul-simpul pada graf dibentuk secara dinamis pada saat runtime. Pembentukan dilakukan secara menyebar (*breadth-first search*).

Untuk masing-masing simpul yang terbentuk, apabila nilai support lebih kecil atau sama dengan minimum supportnya, simpul tersebut dipangkas. Untuk simpul-simpul yang tersedia, kombinasikan dengan simpul lainnya untuk memperoleh itemset level berikutnya. Dan seterusnya dilakukan hingga tidak terdapat pola itemset yang dapat dikombinasikan lagi. Berikut merupakan langkah-langkahnya

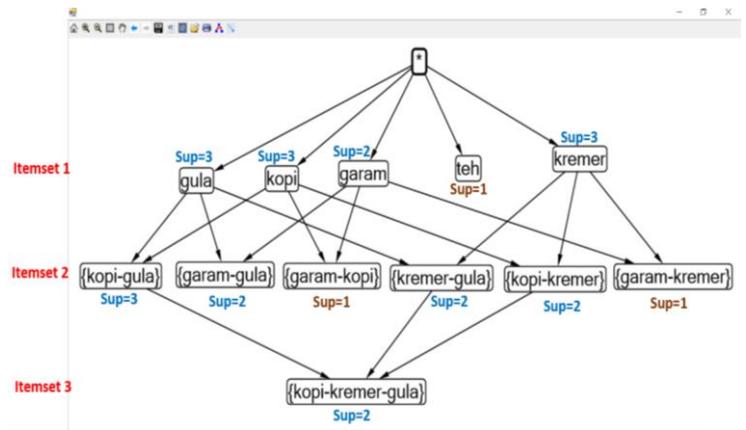


Fig 5: pembentukan itemset secara BFS dengan min\_sup=2  
Sup: nilai support  
Min\_sup: nilai minimum support  
(Sumber: Penulis)

Dari proses di atas, ambil semua simpul dengan nilai lebih dari atau sama dengan min\_sup yang tidak memiliki derajat keluar. Subset dari simpul-simpul itu juga merupakan *frequent pattern mining*.

Kemudian diperoleh hasil itemset yang sering dibeli bersamaan berikut:

Item	Support	Persentase support
{kopi, kremer, gula}	2	0.167
{garam, gula}	2	0.167
{kopi}	3	0.25
{kopi, kremer}	2	0.167
{kopi, gula}	3	0.25
{kremer}	3	0.25
{kremer, gula}	2	0.167
{garam}	2	0.167
{gula}	3	0.25

Tabel 4: Hasil frequent pattern mining

Untuk jumlah data yang lebih banyak, hasil analisis ini akan lebih akurat, kemudian dapat digunakan untuk memprediksi minat pembeli untuk membeli item yang berkorelasi kuat dengan item yang diminatinya. Strategi ini dapat diterapkan pada marketing untuk memasang promosi pada customer yang tepat.

### B. Performansi Pattern Mining dengan Algoritma Apriori

Dalam implementasinya, Algoritma Apriori melakukan pemangkasan (prune) jumlah kandidat yang harus dihitung nilai support-nya. Pemangkasan inilah yang membuat Algoritma Apriori memiliki performa yang baik.

Namun Algoritma ini juga memiliki kekurangan, karena pada strategi ini proses scanning wajib dilakukan pada setiap kali iterasi sehingga akan memerlukan waktu yang cukup lama dan kemampuan komputasi yang besar.

## V. SIMPULAN

*Frequent subgraph mining* merupakan salah satu operasi graph mining yang digunakan untuk mengekstraksi pola yang muncul dari data populasi. Metodi ini dapat diimplementasikan dengan Algoritma Apriori untuk melakukan pemindaian pola *frequent subgraph*. Algoritma ini menggunakan pendekatan *breadth-first search*. Metode pencarian pola ini dapat dimanfaatkan pada analisis marketing. Penentuan pola-pola menarik dari database penjualan dapat menghasilkan informasi yang berguna untuk melakukan strategi bisnisselanjutnya. Misalnya dengan mencari pola produk-produk yang saling berkorelasi kuat dapat menghasilkan prediksi minat pembeli untuk membeli produk-produk yang berkaitan dengan produk yang diminatinya, sehingga mendorong penjualan produk yang lebih besar.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah memberikan petunjuk dan kelancaran dalam menulis makalah ini. Penulis juga berterima kasih kepada kedua orangtua Penulis yang telah mendukung Penulis hingga sejauh ini Penulis berkuliah di Teknik Informatika. Tak lupa Penulis ucapkan terima kasih kepada Bapak Dr.Ir. Rinaldi Munir, MT. yang telah memberikan tantangan bagi Penulis untuk mengembangkan kemampuan pemecahan masalah kehidupan nyata dengan Informatika dan menuangkannya kepada karya ilmiah. Diharapkan semoga karya ilmiah ini dapat bermanfaat bagi sesama dalam menambah pengetahuan dan mengembangkan keilmuan Data Mining.

## REFERENSI

- [1] Munir, Rinaldi. (2006). Matematika Diskrit, Bandung: Penerbit Informatika Bandung.
- [2] Munir, Rinaldi. (2018). Diktat Kuliah IF2211. Strategi Algoritma. Program Studi Teknik Informatika. Institut Teknologi Bandung
- [3] Jiawei Han, Jian Pei, Micheline Kamber. (2006). Data Mining, Southeast Asia Edition, Second Edition. Univerity of Illinois at Urbana-Champaign. Illinois, U.S.
- [4] Viger, Phillippe Fournier, An Introduction to Frequent Pattern Mining, <http://data-mining.philippe-fournier-viger.com/introduction-frequent-pattern-mining/>, 2013, diakses 13 Mei 2018.
- [5] Ramasamy, Vijayalakshmi & Nadarajan, R & Roddick, John & Thilaga, M & Nirmala, and. (2011). FP-GraphMiner - A Fast Frequent Pattern Mining Algorithm for Network Graphs. Journal of Graph Algorithms and Applications. 15. 753-776. 10.7155/jgaa.00247.
- [6] Henderson, Tim. (2017). Frequent Subgraph Analysis and its Software Engineering Applications. <https://hackthology.com/frequent-subgraph-analysis-and-its-software-engineering-applications.html>, diakses 13 Mei 2018.
- [7] Efficient Mining for Large-Scale Graph. [https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201312ra1\\_s.html](https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201312ra1_s.html).
- [8] Data Structures and Breadth First Traversal , [https://www.tutorialspoint.com/data\\_structures.../breadth\\_first\\_traversal.html](https://www.tutorialspoint.com/data_structures.../breadth_first_traversal.html), diakses 13 Mei 2018.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 13 Mei 2018



Erma Safira Nurmaryita (13516072)