Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II
TINJAUAN PUSTAKA

2.1 Market Basket Analysis

Market Basket Analysis (MBA) merupakan sebuah metode dalam data mining yang digunakan untuk mengamati dan mempelajari kebiasaan konsumen dalam berbelanja dengan menemukan asosiasi dan hubungan terhadap beberapa produk yang tersedia dengan sumber data riwayat konsumen serta kebiasaan dalam transaksi (Olson & Delen, 2008). Teknik ini sering disebut *assosiation rule analysis* yang merupakan salah satu cara untuk melakukan *data mining* (Olson & Shi, 2007).

Cara kerja aplikasi *market basket analysis* yang umumnya dikenal sering menggunakan *trend* pembelian yang sama yakni produk A yang biasanya juga diikuti oleh pembelian produk B atau produk C. Informasi dan data ini digunakan oleh para pemilik data dalam strategi pemasaran pemilik suatu retail tersebut. Salah satunya adalah dengan menentukan susunan rak suatu produk yang menyebabkan lokasi produk A, produk B dan produk C berada bersebelahan dan meningkatkan penjualan terhadap ketiga produk tersebut secara langsung.

Dengan menganalisis data transaksi dalam suatu *range* tertentu dengan batasan-batasan yang ditentukan, maka *trend* dan pola pembelian sebuah produk dapat ditemukan dengan teliti. Informasi tersebut kemudian dapat digunakan untuk membantu pemilik data untuk meningkatkan total penjualan melalui promosi produk yang bervariasi dan pembuatan paket produk yang sesuai dengan minat pembeli.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2.2 Machine Learning

Machine learning adalah suatu ilmu yang berhubungan dengan kepintaran buatan (Mitchell, 1977). Dalam buku *machine learning*, Tom Mitchell menjelaskan tentang proses bagaimana cara kerja dan cara membangun sebuah program pada komputer yang dapat menganalisa dan mempelajari sebuah kondisi berdasarkan pengalaman atau data lampau. Hasil analisa data yang digunakan sebagai bahan pembelajaran dapat diperoleh dengan beberapa cara di antaranya melalui perhitungan menggunakan rumus-rumus sesuai dengan tujuan algoritma masing-masing.

2.3 Decission Support System

Decission Support System (DSS) merupakan sistem pendukung keputusan yang interaktif, sistem berbasis komputer yang membantu pengguna dalam kegiatan penilaian dan pemilihan. DSS sangat membantu dalam situasi dimana jumlah informasi yang tersedia sangat banyak yang menyebabkan terhambatnya pembuatan keputusan. DSS juga dapat membantu kekurangan kognitif manusia dengan mengintegrasikan berbagai sumber informasi, menyediakan akses untuk pengetahuan yang relevan, dan membantu proses penataan keputusan (Drudzel, 2002).

2.4 Data Mining

Data mining adalah proses mencari dan menggali pola baru dari sebuah data set data yang besar dan tidak terstruktur dan melibatkan metoda-metoda dari ilmu statistika, intelegensia buatan, dan juga manajemen *database* (Fayyad dkk, 1996).

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

Tugas utama *data mining* adalah melakukan analisis terhadap set-set data dengan tujuan untuk mencari pola yang menarik dan penyajian kepada *end-user* melalui penggambaran secara visual yang sebelumnya tidak diketahui sebagai sebuah potensi lainnya, seperti dependensi (*association rule mining*), mencari data yang tidak umum (*anomaly detection*), dan pengelompokan data (*cluster analysis*) (Fayyad dkk, 1996). Pola ini kemudian dipahami sebagai kesimpulan dari masukkan data dan dapat dipergunakan dalam penelitian lebih lanjut. Yang termasuk dalam *data mining* di antaranya adalah pengumpulan data, persiapan data, interpretasi hasil dan pelaporan.

Algoritma dalam *data mining* sangat variatif dan berkembang dengan cepat dengan hasil performa yang berbeda-beda pula. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses data secara keseluruhan serta jenis data yang hendak diproses. Karakteristik dari *data mining* adalah *volume* data yang begitu besar dan harus dianalisa dengan teknik yang otomatis, *noisy*, *incomplete* data, data yang bersifat heterogenous (Chandrawati, 2010:2).

Data mining adalah proses analisis dari sekumpulan data yang diamati (dengan jumlah yang sangat besar) untuk menemukan hubungan / pola yang tidak terduga dan merekap data tersebut dengan cara yang dapat dipahami dan berguna bagi pemilik data (Chandrawati, 2010). *Data mining* membantu perusahaan untuk mendapatkan suatu pola dan *knowledge* dari data-data yang tersimpan di dalam basis data perusahaan yang berisi *history* transaksi perusahaan tersebut yang telah lama tertumpuk dan terolah dengan baik. Pengetahuan yang diperoleh tersebut akan menjadi acuan pedoman dalam mengambil tindakan bisnis sebagai upaya

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

pemeliharaan dan peningkatkan tingkat kompetitif bisnis perusahaan (Yusuf , 2006:17).

2.4.1 Metode dalam Data Mining

Larose (2005) dalam bukunya yang berjudul "*Discovering Knowledge in Data: An Introduction to Data Mining*", *data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu sebagai berikut.

1) Deskripsi

   Suatu kondisi analis dan peneliti mencoba mencari cara untuk merepresentasikan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari suatu pola kecenderungan sering memberikan penjelasan untuk suatu pola atau kecenderungan.

2) Klasifikasi

   Klasifikasi digunakan untuk memprediksikan kelas data yang bersifat *categorical*. Dalam klasifikasi, terdapat target variabel kategori seperti contohnya penggolongan usia yang dapat dipisahkan dalam empat kategori, yaitu anak-anak, remaja, dewasa, dan manula. Contohnya, model klasifikasi yang dibangun untuk mengkategorisasikan aplikasi-aplikasi bank sebagai aplikasi yang aman atau beresiko.

3) Estimasi

   Estimasi memiliki definisi yang hampir mirip dengan klasifikasi. Yang membedakannya adalah variabel target estimasi lebih terfokus pada angka daripada ke arah kategorikal. Model disusun menggunakan baris data lengkap yang menyediakan *value* dari variabel target sebagai nilai acuan

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

prediksi. Selanjutnya, pada peninjaunan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

4) Prediksi

Prediksi memiliki definisi yang sama dengan klasifikasi dan estimasi, tetapi dalam prediksi *value* dari suatu hasil akan tersedia di masa mendatang. Beberapa metode yang diaplikasikan dalam klasifikasi dan estimasi dapat digunakan untuk prediksi juga. Contohnya, model yang dibangun untuk memprediksikan kebiasaan berbelanja konsumen berdasarkan pendapatan dan pekerjaannya.

5) Pengklusteran(*Clustering*)

Metode *clustering* bertujuan untuk memaksimalkan persamaan dalam satu *cluster* dan meminimalkan perbedaan antar *cluster*. Metodenya dengan mengelompokkan data, *record*, pengamatan, atau memperhatikan untuk membentuk kelas-kelas objek baru yang memiliki kemiripan satu dengan yang lainnya. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar. Contoh lainnya adalah untuk tujuan audit akuntansi,

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.

6) Asosiasi

Tugas asosiasi dalam *data mining* adalah untuk mencari dan menemukan atribut yang muncul dalam satu waktu berdasarkan beberapa transaksi sejenis terdahap suatu *item* dengan *item* lainnya. Metodenya dengan menemukan atribut yang muncul dalam satu waktu. Metode aturan asosiasi membuat aturan berdasarkan kondisi yang sering terjadi. Aturan yang dibuat untuk data dalam kondisi X, diharapkan dapat diterapkan untuk data-data lainnya dalam kondisi Y. Contoh metode asosiasi : Meneliti jumlah pelanggan dari suatu perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan. Contoh lainnya adalah menemukan barang dalam *supermarket* yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.4.2 Komponen Algoritma dalam Data Mining

Setelah mengenal beberapa tipe metode umum dalam *data mining*, penyusunan algoritma yang lebih spesifik dapat dimulai. Menurut Fayyad dkk (1996), garis besar komponan utama dalam algoritma *data mining* sebagai berikut.

1. Model *Representation*. Representasi model adalah istilah yang digunakan untuk menjelaskan pola yang akan ditemukan. Apabila representasi model ini terlalu terbatas maka sistem yang dibangun nantinya tidak akan bisa memproduksi hasil yang akurat berdasarkan data yang ada. Seorang analis data yang baik harus bisa memahami asumsi-asumsi representasional yang

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

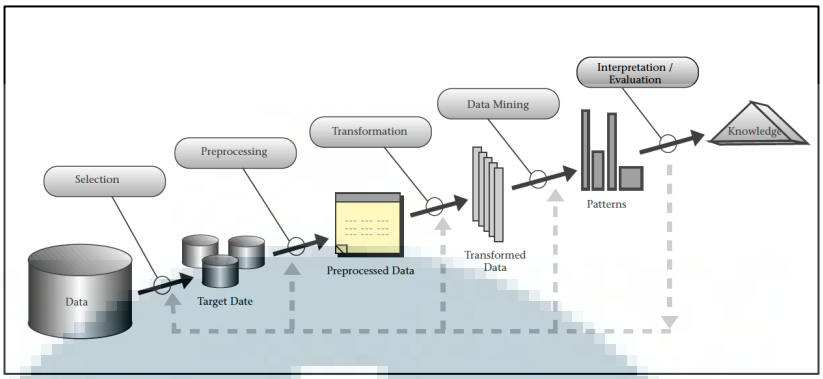
mungkin terkait dengan metode-metode tertentu. Hal ini sama pentingnya bagi perancang algoritma untuk memahami asumsi-asumsi representasional yang dibuat oleh algoritma-algoritma tertentu.

2. Model *evaluation*. Evaluasi model adalah nilai kuantitatif dari seberapa baik sebuah pola (metode dan parameternya) memenuhi tujuan dari proses *knowledge discovery*. Sebagai contoh, metode prediktif sering diuji tingkat akurasi prediksinya secara empiris dengan beberapa set percobaan.
3. Metode *search* terdiri dari dua komponen yaitu parameter *search* dan model *search*. Setelah kriteria model *representation* dan model *evaluation* ditetapkan, maka persoalan data *mining* yang tersisa hanyalah persoalan optimalisasi. Dalam parameter *search*, algoritma harus mencari parameter-parameter yang mengoptimalkan kriteria model *evaluation* yang diberikan terhadap data yang diamati dan model *representation* yang telah ditetapkan. Model *search* muncul sebagai sebuah perulangan (*loop*) atas parameter *search*.

2.4.3 Tahapan dalam Data Mining

Proses *Knowledge Discovery in Databases* (KDD) menurut Fayyad dll(1996), dalam jurnalnya yang berjudul "*From Data Mining to Knowledge Discovery in Databases*", dapat diilustrasikan dengan Gambar 2.1 (Fayyad dkk, 1996).

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016



Gambar 2.1 Tahapan Penyusunan Knowledge Discovery in Databases

Tahapan penyusunan KDD seperti yang digambarkan pada Gambar 2.1 umumnya didefinisikan sebagai berikut (Chandrawati, 2010:2).

1) *Selection*: yaitu proses pemilihan dan penyaringan data berdasarkan beberapa kriteria sesuai kebutuhan peneliti, misalkan berdasarkan jenis barang.
2) *Pre-processing*: yaitu mempersiapkan data dengan cara membersihkan data yang akan diproses serta menghapus informasi atau *field-field* yang tidak dibutuhkan, yang nantinya akan memperlambat kinerja proses *query*.
3) *Transformation*: yaitu proses mentransformasikan data ke bentuk yang lebih *navigable* dan *useable*.
4) Data *mining*: pada bagian ini, dipusatkan untuk mendapatkan pola-pola dari data. Setidaknya ada 3 teknik yang paling sering digunakan pada data *mining*, yaitu: *classification, clustering* dan *association rule mining*.
5) *Interpretation and evaluation*: dalam proses ini *pattern* atau pola-pola yang telah diidentifikasi oleh sistem kemudian diterjemahkan ke dalam bentuk *knowledge* yang lebih mudah dimengerti oleh *end-user* untuk mengambil keputusan.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2.5 Association Rule

Suatu metode *data mining* yang digunakan mencari sekumpulan *item* yang sering muncul bersamaan di antara sebuah *set items* atau objek pada transaksi di dalam basis data (Han,2004). Umumnya aturan *association* ini dianalogikan dengan keranjang belanjaan setiap konsumen pada *supermarket*. Dari keranjang belanjaan para pengunjung *supermarket* akan dapat diketahui, barang apa saja yang sering dibeli bersamaan dan barang mana saja yang tidak. *Association rule* umumnya mengambil bentuk *IF-THEN* yang menggabungkan beberapa *items* menjadi satu, misalnya: *IF* A *and* B *THEN* C (Han,2004). *IF* pada *association rules* dianalogikan dengan sisi *antecedent* dan *THEN* dianalogikan dengan sisi *consequent*.

Sebagai contoh, sebuah transaksi I = {i1,i2,i3,…,id} adalah himpunan *item* / produk dari transaksi, sedangkan T = {t1,t2,t3,…,tn} adalah himpunan *order* transaksi. Setiap transaksi t1 terdiri dari *item* yang merupakan bagian dari I. *Support* dari sebuah *itemset* X disimbolkan dengan sup(X). *Confidence* dari sebuah *itemset* X disimbolkan dengan conf(X). Untuk menemukan *association rule* dari sebuah transaksi diperlukan nilai batasan yaitu *minimum support* dan *minimum confidence* (Agrawal,1993).

Secara teori, beberapa hal yang digunakan untuk mengukur apakah sekumpulan *items* sering muncul bersamaan atau tidak adalah dengan menggunakan parameter *support*, *confidence*, dan beberapa *rule selection methods* seperti *lift ratio*.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2.6 Aturan Acuan Ukur Association Rule

Hasil rekomendasi yang dihasilkan dari *data mining* dengan metode *association rule* dapat diukur dengan parameter/ acuan ukur sebagai berikut.

2.6.1 Support

Support adalah persentase dari seluruh transaksi yang terjadi yang mengandung *item set* tersebut. Sebagai contoh, produk susu dan telur muncul secara bersamaan dalam 50 transaksi dari total 500 transaksi sehingga nilai *support* dari susu dan telur dalam transaksi tersebut adalah 10% ((50/500)*100%). Adapun rumus yang digunakan untuk menghitung nilai *support* adalah seperti Rumus 2.1 (Larose, 2005).

$$Support\ (A, C) = \left(\frac{|A,C|}{|T|}\right) * 100\% \qquad ...(2.1)$$

Keterangan :

|A,C| = Jumlah kemunculan A,C dalam seluruh transaksi

|T| = Jumlah transaksi yang terjadi

2.6.2 Confidence

Confidence adalah perbandingan antara nilai *support* dari himpunan *items* yang mendahuluinya dengan nilai *support* dari himpunan *items* yang terdapat di dalam *rule* tersebut. *Confidence* juga dapat diartikan sebagai suatu nilai kepastian yaitu kuatnya hubungan antar *item* dalam sebuah *itemset. Confidence* dapat dicari setelah pola frekuensi munculnya sebuah *item* ditemukan. Sebagai contoh, apabila

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

terdapat *rule* "Jika membeli roti dan keju maka konsumen membeli susu". Untuk menghitung nilai *confidence* tersebut digunakan Rumus 2.2 (Larose, 2005).

$$Confidence\ (R) = \left(\frac{Support\ (\mathrm{A,C})}{Support\ (\mathrm{A})}\right) * 100\% \qquad ...(2.2)$$

2.6.3 Lift Ratio

Lift ratio digunakan untuk mengevaluasi kuat tidaknya sebuah aturan asosiasi dengan cara membandingkan antara nilai *support* dari *antecedent* dan *consequent* dibagi nilai *support antecedent* dikalikan nilai *support consequent* (Han, 2004). Penerapan *lift ratio* pada data transaksi *supermarket* dapat dianalogikan untuk mencari kombinasi produk susu dan kantong belanja yang seharusnya tidak perlu dikombinasikan karena kantong belanja bukan merupakan produk yang diperlukan untuk kombinasi tersebut. Untuk mencari nilai *lift ratio*, digunakan yaitu Rumus 2.3.

$$Lift\ Ratio = \frac{Support\ (A,C)}{Support\ (A) * Support\ (C)} \qquad ...(2.3)$$

Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya (Han, 2004). Penggunaan *lift ratio* dapat membantu sebagai acuan ukur dengan beberapa alasan sebagai berikut (Han, 2004).

1) Jika suatu *rule* memiliki *confidence* tinggi, berarti *rule* tersebut mencirikan aturan asosiasi yang kuat.
2) Pemilihan *rule* berdasar *confidence* bisa menipu, sebab jika *antecedent(ant)*/*consequent(cons)* memiliki *support* yang tinggi, maka *rule* dapat memiliki *confidence* tinggi, walaupun sebetulnya *independent*.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

Berikut adalah acuan nilai *rules* dalam pengukuran terhadap *lift ratio*(Han, 2004).

1) Sebuah *lift ratio* > 1.0 menyatakan *antecedent* dan *consequent* muncul lebih sering dari yang diharapkan. Kemunculan *antecedent* memiliki efek positif terhadap kemunculan *consequent.*
2) Sebuah *lift ratio* < 1.0 menyatakan *antecedent* dan *consequent* muncul lebih jarang dari yang diharapkan. Kemunculan *antecedent* memiliki efek negatif terhadap kemunculan *consequent.*
3) Sebuah *lift ratio* ≈ 1.0 menyatakan *antecedent* dan *consequent* muncul hampir selalu bersamaan seperti yang diharapkan. Kemunculan *antecedent* hampir tidak memiliki pengaruh terhadap kemunculan *consequent.*

2.7 Algoritma Frequent Pattern Growth (FP-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 (Han, 2004). FP-Growth menggunakan pendekatan yang berbeda dari paradigma yang selama ini sering digunakan, yaitu paradigma apriori.

Paradigma apriori yang dikembangkan oleh Agrawal dan Srikan (1994), yaitu *anti-monotone Apriori Heuristic:* Setiap pola dengan panjang pola k yang tidak sering muncul (tidak *frequent*) dalam sebuah kumpulan data, maka pola dengan panjang (k+1) yang mengandung sub-pola k tersebut tidak akan sering muncul pula (tidak *frequent*).

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

Ide dasar paradigma apriori ini adalah mencari himpunan peserta / himpunan *item* dengan panjang (k+1) dari sekumpulan pola *frequent* dengan panjang k, lalu mencocokkan jumlah kemunculan pola tersebut dengan informasi yang terdapat dalam *database* (Samuel, 2007). Adapun hal ini akan mengakibatkan algoritma apriori akan melakukan *scanning database* yang berulang-ulang, apalagi jika jumlah data yang besar. Berbeda dengan algoritma FP-Growth yang hanya memerlukan dua kali *scanning database* untuk menentukan *frequent itemset*.

FP-Growth menggunakan struktur FP-Tree untuk menemukan himpunan lengkap dari *frequent pattern*. Algoritma FP-Growth menghasilkan jumlah kandidat yang sedikit dan jumlah pembacaan *dataset* yang minimal sehingga waktu responnya cepat (Han, 2004).

Sebuah FP-Tree terdiri dari *root* yang tidak berisi apa-apa atau *null*, himpunan *item prefix subtree* sebagai anak dari *root* dan sebuah *header* berisi *frequent-item*. Setiap *node* pada *prefix subtree* terdiri dari *node link, count,*dan *item-name*. *Count* merupakan total jumlah transaksi pada *database* yang menjadi bagian dari *prefix* yang diwakilkan oleh *node*, dan *node-link* berfungsi untuk menghubungkan *node* yang satunya ke *node* selanjutnya pada FP-Tree yang memiliki nama *item* yang sama / sejenis. Tabel *header* terdiri dari dua *field* yaitu *head of node-link* dan *item-name*(Purwanti, 2010). FP-Tree hanya berisi beberapa *item* yang *frequent* yang telah diurutkan berdasarkan nilai *support*-nya.

Setelah FP-Tree selesai terbentuk dengan berisi data-data transaksi beserta *item* terkait, algoritma FP-Growth dijalankan secara rekursif untuk membangun *conditional pattern base* dan *conditional* FP-Tree untuk setiap *frequent item* yang

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

selanjutnya digunakan untuk men-*generate* semua *frequent itemset.* Gambar 2.2 adalah *pseudocode*-nya.

```
Input:      A database DB, represented by FP-tree constructed
            according to Algorithm 1, and a minimum support
            threshold ξ .
Output:     The complete set of frequent patterns
Method:     call FP-growth(FP-tree, null).

Procedure FP-growth(Tree, α)
{
(1)  if Tree contains a single prefix path // Mining single prefix-path
     FP- tree
(2)  then {
(3)  let P be the single prefix-path part of Tree;
(4)  let Q be the multipath part with the top branching node replaced by
     a null root;
(5)  for each combination (denoted as β) of the nodes in the path P do
(6)  generate pattern β ∪ α with support = minimum support of nodes
     in β;
(7)  let freq pattern set(P) be the set of patterns so generated; }
(8)  else let Q be Tree;
(9)  for each item ai in Q do { // Mining multipath FP-tree
(10) generate pattern β = ai ∪ α with support = ai .support;
(11) construct β's conditional pattern-base and then β's conditional FP-
     tree Treeβ ;
(12) if Treeβ = ∅
(13) then call FP-growth(Treeβ , β);
(14) let freq pattern set(Q) be the set of patterns so generated; }
(15) return(freq pattern set(P) ∪ freq pattern set(Q) ∪ (freq pattern
     set(P) × freq pattern set(Q)))
}
```

Gambar 2.2 Pseudocode FP-Growth (Han,2004)

2.7.1 Frequent Pattern Tree (FP-Tree)

FP-Tree merupakan struktur penyimpanan data yang *compact* berbentuk pohon struktural (Han, 2004). FP-Tree dibangun melalui pemetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam FP-Tree. Kelebihan dari FP-Tree yaitu terdapat pada struktur penyimpanan data transaksi dengan tingkat efisiensi

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

yang tinggi karena hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien untuk membangun struktur *tree*. Dengan menggunakan struktur *tree* algoritma FP-Growth dapat langsung mengekstrak *frequent itemset* dari susunan FP –Tree yang telah terbentuk (Han, 2004).

Misalkan diberikan tabel data transaksi seperti Tabel 2.1, dengan *minimum support* =0.4 dari total transaksi (*minimum support* = 2).

Tabel 2.1 Tabel Data Transaksi Mentah

Transaksi	Item
1	Gula, Kopi, Teh
2	Gula, Kopi, Teh
3	Gula, Susu, Roti
4	Roti, Gula, Air
5	Gula, Susu, Kopi

Frekuensi kemunculan tiap item dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel Frekuensi kemunculan tiap karakter

Rank	Produk	Frekuensi
1	Gula	5
2	Kopi	3
3	Teh	2
4	Roti	2
5	Susu	2
6	Air	1

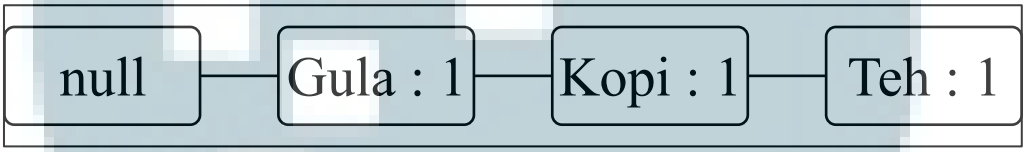
Setelah dilakukan *scanning* pertama, didapat *item* yang memiliki frekuensi di atas *minimum support* = 2 adalah gula,kopi,teh, dan roti. *Item* tersebut yang akan berpengaruh dan akan dimasukkan ke dalam FP-Tree, selebihnya (air) akan dibuang karena tidak berpengaruh signifikan (di bawah angka *minimum support*). Tabel 2.3 mendata jumlah kemunculan *item* yang *frequent* dalam setiap transaksi dan diurutkan berdasarkan yang frekuensinya paling tinggi hingga paling rendah.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

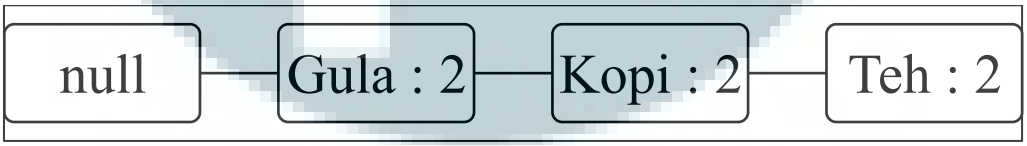
Tabel 2.3 Tabel Data Transaksi 1

Transaksi	Item
1	Gula, Kopi, Teh
2	Gula, Kopi, Teh
3	Gula, Roti, Susu
4	Gula, Roti
5	Gula, Kopi, Susu

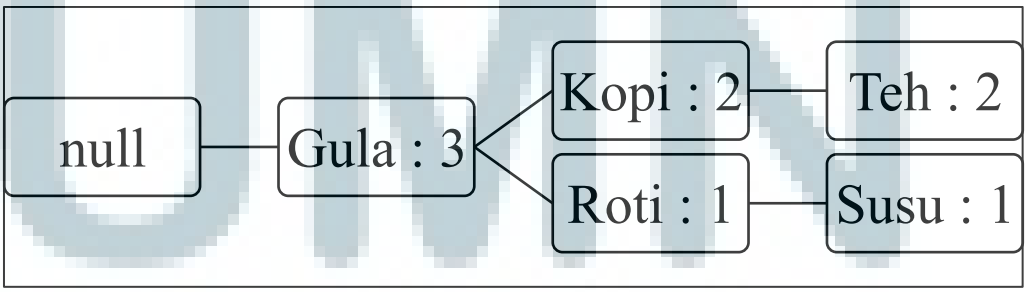
Tabel 2.3 adalah contoh/ gambaran mengenai penyusunan transaksi yang telah terbentuk dan tersusun sesuai prosedur untuk kemudian dimasukkan ke FP-Tree.



Gambar 2.3 Hasil Pembentukan FP–Tree Pembacaaan Transaksi Pertama

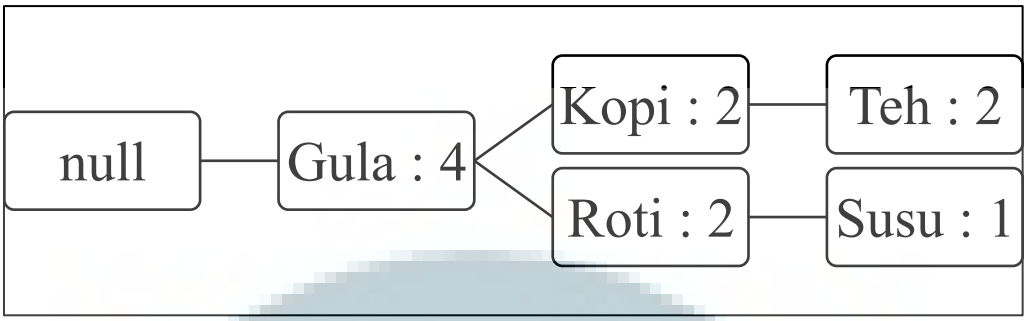


Gambar 2.4 Hasil Pembentukan FP–Tree Pembacaaan Transaksi Kedua

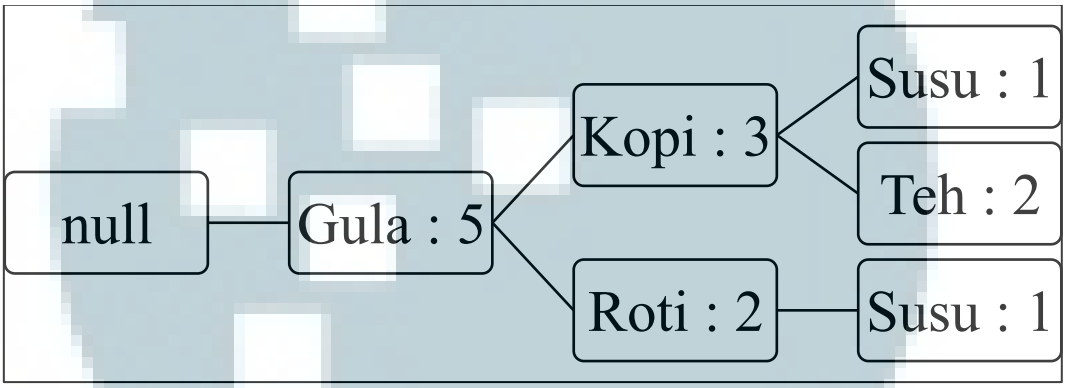


Gambar 2.5 Hasil Pembentukan FP–Tree Pembacaaan Transaksi Ketiga

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016



Gambar 2.6 Hasil Pembentukan FP–Tree Pembacaaan Transaksi Keempat



Gambar 2.7 Hasil Pembentukan FP–Tree Pembacaaan Transaksi Kelima

Gambar 2.3 sampai Gambar 2.7 menunjukkan proses terbentuknya FP–Tree pada pembacaan transaksi pertama hingga selesai. Setiap simpul pada FP-Tree mengandung nama sebuah *item* dan *counter support* yang berfungsi untuk menghitung frekuensi kemunculan *item* tersebut dalam tiap lintasan *tree*. Pembentukan FP-Tree merepresentasikan data transaksi 1 pada Tabel 2.3 dibentuk dengan cara sebagai berikut.

1) Kumpulan data di-*scan* pertama kali untuk menentukan *support count* / frekuensi dari setiap *item*. *Item* yang tidak *frequent* (tidak memenuhi *minumum support*) akan dibuang, sedangkan *frequent item* akan dimasukkan dan disusun dengan urutan menurun ke bawah pada struktur *tree* tersebut.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2) *Scanning* kedua, yaitu pembacaan transaksi pertama {gula, kopi, teh} yang akan membuat simpul gula, kopi dan teh, sehingga terbentuk lintasan transaksi *null*→gula→kopi→teh. *Support count* dari setiap simpul bernilai awal 1 karena merupakan *input*-an pertama sebelum transaksi lainnya dimasukkan pada struktur itu (contoh: *null*→gula→kopi→teh kembali, maka gula, kopi dan teh masing-masing akan bernilai 2 karena telah tertimpa).

3) Transaksi kedua memiliki *prefix* transaksi yang sama dengan transaksi pertama, yaitu {gula}, maka lintasan transaksi kedua dapat ditimpakan di {gula}, sambil menambah *support count* dari {gula}, dan selanjutnya menimpa kembali karena memiliki *item* yang sama. Hal lainnya adalah membuat lintasan baru apabila setelah {gula} bukan *item* {kopi}.

4) Setelah pembacaan transaksi kedua {gula,kopi,teh}, terbentuk lintasan kedua yaitu *null*→gula→roti→susu pada struktur *tree* tersebut. *Support count* masing-masing *count* juga bernilai awal 1 (kecuali yang sudah ditimpa).

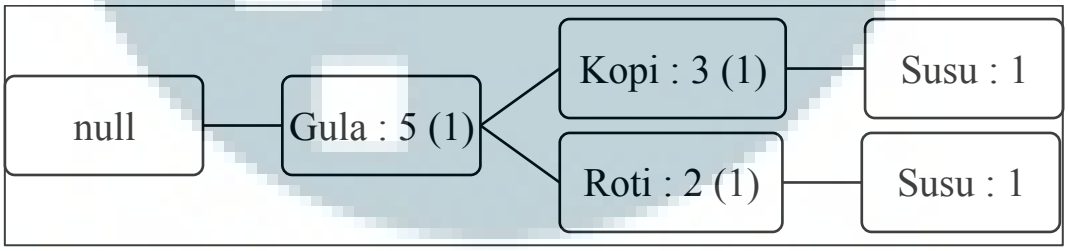
Proses ini dilanjutkan sampai FP-Tree berhasil dibangun berdasarkan tabel data transaksi yang diberikan. Setelah tahap pembangunan FP-Tree dari sekumpulan data transaksi, akan diterapkan algoritma FP-Growth untuk mencari *frequent itemset* yang signifikan. Algoritma FP-Growth dibagi menjadi tiga langkah utama dengan proses yang berurutan, yaitu sebagai berikut (Han, 2004).

1) Tahap pembangunan *conditional pattern base*
2) Tahap pembangkitan *conditional* FP-Tree
3) Tahap pencarian *frequent itemset*

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2.7.2 Tahap Pembangunan Conditional Pattern Base

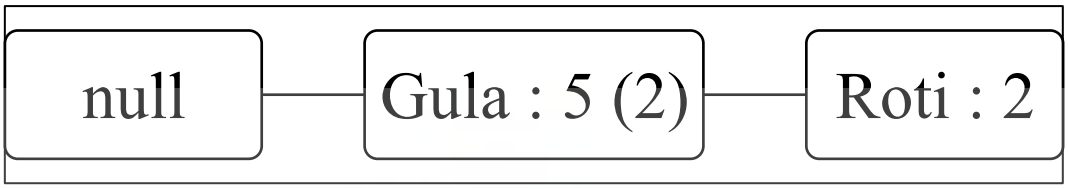
Conditional Pattern Base merupakan *sub-database* yang berisi *prefix path* (lintasan/ jalur *prefix*) dan *suffix pattern* (pola akhiran) (Han, 2004). Pembangkitan *conditional pattern base* didapatkan melalui penyimpanan struktur *tree* melalui pendekatan FP-Tree yang telah dibangun dan disusun sesuai prosedur sebelumnya. Untuk menemukan *conditional pattern base* dari contoh kasus di atas, maka perlu ditentukan *tree* dengan *path y*ang berakhir dengan *support count* terkecil, dalam contoh kasus berikut yaitu *item* susu. Berturut-turut ditentukan juga *path* yang berakhiran di roti, teh, kopi, dan gula. Proses pembentukan dapat dilihat pada Gambar 2.8.



Gambar 2.8 Lintasan yang Mengandung Item Susu

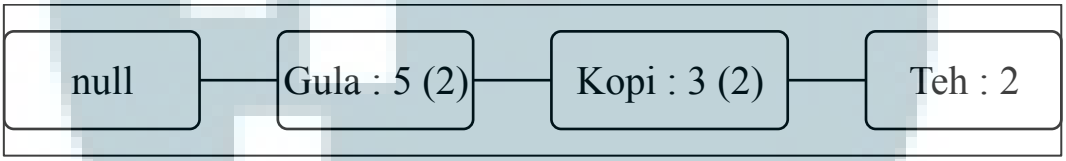
Gambar 2.8 merupakan lintasan *conditional pattern base* dengan akhiran lintasan yang mengandung *item* susu berdasarkan FP-Tree yang telah dihasilkan. Kemudian nilai *support item* yang terdapat pada lintasan tersebut akan disesuaikan dengan nilai *support* pada *item* susu yakni bernilai 1.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016



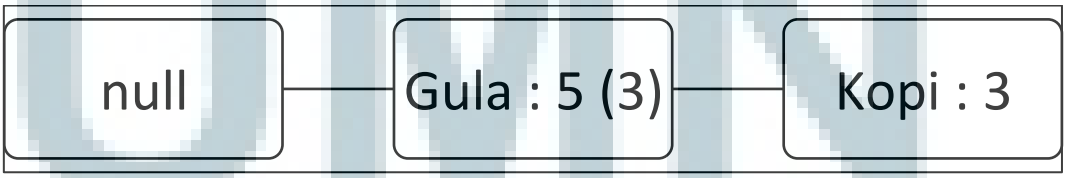
Gambar 2.9 Lintasan yang Mengandung Item Roti

Gambar 2.9 merupakan lintasan *conditional pattern base* dengan akhiran lintasan yang mengandung *item* roti berdasarkan FP-Tree yang telah dihasilkan. Kemudian nilai *support item* yang terdapat pada lintasan tersebut akan disesuaikan dengan nilai *support* pada *item* roti yakni bernilai 2.



Gambar 2.10 Lintasan yang Mengandung Item Teh

Gambar 2.10 merupakan lintasan *conditional pattern base* dengan akhiran lintasan yang mengandung *item* teh berdasarkan FP-Tree yang telah dihasilkan. Kemudian nilai *support item* yang terdapat pada lintasan tersebut akan disesuaikan dengan nilai *support* pada *item* teh yakni bernilai 2.



Gambar 2.11 Lintasan yang Mengandung Item Kopi

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

Gambar 2.11 merupakan lintasan *conditional pattern base* dengan akhiran lintasan yang mengandung *item* kopi berdasarkan FP-Tree yang telah dihasilkan. Kemudian nilai *support item* yang terdapat pada lintasan tersebut akan disesuaikan dengan nilai *support* pada *item* kopi yakni bernilai 3. Dengan demikian, didapatkan tabel *conditional pattern base* dari contoh kasus transaksi 1 dalam Tabel 2.4.

Tabel 2.4 Contoh Tabel Conditional Pattern Base

Item	**Conditional Pattern Base**
Susu	{{Gula, Kopi : 1}, {Gula, Roti : 1}}
Roti	{{Gula : 2}}
Teh	{{Gula, Kopi : 2}}
Kopi	{{Gula : 3}}

2.7.3 Tahap Pembangkitan Conditional FP-Tree

Support count dari setiap *item* pada setiap *conditional pattern base* akan dijumlahkan pada tahap ini, kemudian setiap *item* yang memiliki jumlah *support count* lebih besar sama dengan *minimum support count* akan dibangkitkan dengan *conditional* FP-Tree (Han, 2004). Sebagai contoh untuk *conditional* FP-Tree, titik temu untuk produk susu adalah di produk gula dengan nilai *support count* memenuhi *minimum support*. Sehingga didapatkan untuk *item* susu *conditional* FP-Tree nya adalah gula dengan *support count* 2. Hal tersebut berlaku juga terhadap *item* lainnya. Untuk lebih detailnya, dapat dilihat melalui Tabel 2.5.

Tabel 2.5 Contoh Tabel Conditional FP-Tree

Item	**Conditional FP-Tree**
Susu	{{Gula : 2}}
Roti	{{Gula : 2}}
Teh	{{Gula : 2}, {Kopi : 2},{Gula, Kopi : 2}}
Kopi	{{Gula : 3}}

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

2.7.4 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 (Han, 2004). Algoritma FP-Growth menemukan *frequent itemset* yang berakhiran *suffix* tertentu dengan menggunakan metode *divide and conquer* untuk memecah *problem* menjadi *subproblem* yang lebih kecil (Han, 2004). Untuk *frequent pattern itemset* pada contoh kasus di atas, didapatkan hasil sebagai berikut.

Tabel 2.6 Contoh Tabel Frequent Itemset

Item	Frequent Itemset
Teh	{{Gula, Teh : 2}, {Kopi, Teh : 2},{Gula, Kopi, Teh : 2}}
Susu	{{Gula, Susu : 2}}
Roti	{{Gula, Roti : 2}}
Kopi	{{Gula, Kopi : 3}}

2.7.5 Tahap Pembentukan Association Rules

Setelah *frequent pattern itemset* didapatkan, dilakukan perubahan bentuk menjadi *association rules* dengan mengikuti aturan. Pemrosesan *association rules* berdasarkan *frequent pattern itemset* yang telah dihasilkan oleh algoritma FP-Growth. Berdasarkan *frequent pattern itemset* di atas, didapatkan hasil *association rules* sebagai berikut.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

Tabel 2.7 Contoh Hasil Association Rules

Rule Ke-	Antecedent	Consequent	Frekuensi (Antecedent, Consequent	Frekuensi (Antecedent)	Frekuensi (Consequent)	Support (Antecedent)	Support (Consequent)	Support (Antecedent, Consequent)	Confidence (Antecendent, Consequent)	Lift (Antecendent, Consequent)
Tag	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**	**(G)**	**(H)**	**(I)**	**(J)**
Rumus			**Jumlah kemunculan antecedent dan consequent bersamaan**	**Jumlah kemunculan antecedent**	**Jumlah kemunculan consequent**	**D/total transaksi**	**E/total transaksi**	**C/total transaksi**	**C/D**	**H/(F*G)**
1	Gula	Teh	2	5	2	1	0.4	0.4	0.4	1
2	Teh	Gula	2	2	5	0.4	1	0.4	1	1
3	Kopi	Teh	2	2	2	0.4	0.4	0.4	1	2.5
4	Teh	Kopi	2	2	3	0.4	0.6	0.4	1	1.667
5	Gula	Kopi, Teh	2	5	2	1	0.4	0.4	0.4	1
6	Kopi	Gula, Teh	2	3	2	0.6	0.4	0.4	0.667	1.667
7	Teh	Gula, Kopi	2	2	3	0.4	0.6	0.4	1	1.667
8	Gula, Kopi	Teh	2	3	2	0.6	0.4	0.4	0.667	1.667
9	Gula, Teh	Kopi	2	2	3	0.4	0.6	0.4	1	1.667
10	Teh, Kopi	Gula	2	2	5	0.4	1	0.4	1	1
11	Gula	Susu	2	5	2	1	0.4	0.4	0.4	1
12	Susu	Gula	2	2	5	0.4	1	0.4	1	1
13	Gula	Roti	2	5	2	1	0.4	0.4	0.4	1
14	Roti	Gula	2	2	5	0.4	1	0.4	1	1
15	Gula	Kopi	3	5	3	1	0.6	0.6	0.6	1
16	Kopi	Gula	3	3	5	0.6	1	0.6	1	1

Dihasilkan total 16 *rules* yang memenuhi parameter *minimum support* 0.4. Diberikan beberapa parameter tambahan yang akan digunakan sebagai bahan perhitungan *recommended rules* yang dihasilkan.

2.8 Systematic Sampling

Systematic sampling merupakan metode statistik yang melibatkan pemilihan elemen dari kerangka *sampling* yang telah ditentukan. Dengan *sampling* sistematik, setiap *item* k dipilih untuk menghasilkan *sample* berukuran n dari populasi ukuran N. Nilai k dapat juga disebut sebagai siklus *sampling* dan dapat ditentukan dengan Rumus 2.4 (Black, 2004).

$$k = \frac{N}{n} \qquad ...(2.4)$$

Keterangan :

n = ukuran *sample*

N = total keseluruhan populasi *sample*

k = ukuran interval yang diseleksi

Sebagai contoh *systematic sampling*, pemerintah bagian audit ingin mengambil *sample* perusahaan sebesar 5% dari total perusahaan yang terdaftar di Texas. Total perusahaan yang terdaftar di Texas berjumlah 10.000 (N) perusahaan sehingga 5% dari 10.000 adalah 500 perusahaan (n). Interval yang akan digunakan dalam pemilihan ini adalah 10.000 / 500 yaitu 20 (k). Setelah itu, pengambilan *sample* yang pertama ditentukan secara *random* dari nilai 1 sampai k sebagai *starting point.* Sebagai contoh, dari nilai 1 – 20, diambil nilai 5 sebagai *starting point* sehingga perusahaan ke-5 pertama kali diambil sebagai perusahaan

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016

sample, kemudian perusahaan ke-25 (5+20) diambil sebagai *sample* kembali, perusahaan ke-45 diambil sebagai *sample*, dan seterusnya.

Salah satu keuntungan menggunakan *systematic sampling* adalah karena pengambilan *sample* yang rata dari sekelompok populasi data yang memiliki parameter seperti data yang terkumpul secara periodik / waktu tertentu, huruf abjad yang dapat diurutkan, maupun parameter lainnya (Black, 2004). Dalam kasus *market basket analysis,* pemilihan *systematic sampling* dianggap handal karena suatu produk memiliki masa kejayaan masing-masing (faktor iklan, stok barang, dll) yang menyebabkan produk yang ada di tahun ini belum tentu ada di tahun depan sehingga pengambilan *sample* dengan suatu interval yang ditentukan dapat mewakili produk tersebut dalam interval tersebut dengan pengurutan seluruh transaksi berdasarkan waktu.

Implementasi algoritma..., Susanto Ario, FTI UMN, 2016