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

LANDASAN TEORI

2.1 Market Basket Analysis

Ide *Market Basket Analysis* didapat dari gambaran keranjang pasar yang berisi beberapa produk yang dibeli oleh seseorang saat berbelanja di supermarket (Berry dan Linoff, 2004). Setiap keranjang merepresentasikan apa yang dibeli oleh pelanggan dalam satu waktu. Kumpulan data pembelian yang dilakukan oleh semua pelanggan memberikan banyak informasi bagi perusahaan *retail* seperti barang apa saja yang dibeli secara bersamaan dan kapan barang itu dibeli.

Adapun aspek-aspek dalam *Market Basket Analysis* yaitu *one-to-one analysis* yang menyesuaikan perilaku sebuah toko berdasarkan kebiasaan konsumen dan informasi yang diperoleh dari sekelompok konsumen, dan mempelajari pola pembelian yang dilakukan oleh konsumen pada sebuah toko *retail* untuk meningkatkan penjualan (Charlet dan Kumar, 2012). Untuk meningkatkan penjualan dapat dilakukan *up-selling* dan *cross-selling*. *Up-selling* adalah tindakan memotivasi konsumen agar membeli barang yang lebih menguntungkan bagi penjual seperti menawarkan produk yg lebih baik dibandingkan dengan produk yang lain meskipun harga barang relatif lebih mahal. *Cross-selling* adalah tindakan menawarkan produk lainnya yang bersangkutan dengan produk yang akan dibeli oleh konsumen untuk meningkatkan penjualan.

Contoh penggunaan *Market Basket Analysis* adalah jika pelangggan membeli tepung dan gula, biasanya juga membeli telur dikarenakan kemungkinan pelanggan ingin membuat kue. Informasi tersebut dapat digunakan bagi perusahaan *retail*

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

untuk menaruh rak telur berdekatan dengan rak tepung dan gula sehingga pelanggan tidak kesulitan membeli telur saat berbelanja bahan-bahan kue. Selain penyusunan rak yang berdekatan, berdasarkan frekuensi item yang sering dibeli secara bersamaan dapat digunakan dalam pembuatan *package bundling* sebagai promosi sehingga diharapkan dapat meningkatkan penjualan *cross-selling*.

Market Basket Analysis merupakan salah satu metode dalam *data mining* yang berfokus untuk menemukan *pattern* atau pola pembelian dengan cara mengekstrak perilaku atau kebiasaan dari data transaksi sebuah toko (Charlet dan Kumar, 2012). *Data mining* sendiri merupakan sebagian proses dari proses yang lebih besar yaitu *Knowledge Discovery in Database*.

2.2 Knowledge Discovery in Databases

Knowledge adalah pengetahuan yang awalnya disimpan di dalam buku, tetapi di zaman elektronik, pengetahuan disimpan dalam database. *Knowledge Discovery* adalah gabungan dari beberapa algoritma, kecerdasan buatan, teknik matematika, dan statistik yang berfungsi untuk memprediksi dan menjelaskan pengetahuan baru yang tersembunyi dalam kumpulan data (Fayyed dkk., 1996).

Knowledge Discovery in Databases adalah proses pencarian otomatis untuk mencari *pattern* pada kumpulan data yang dapat dianggap sebagai pengetahuan. Tujuan utama dari KDD adalah untuk mengubah kumpulan data menjadi pengetahuan (Kumar, 2011). Sebagai contoh, pengetahuan yang didapat dengan menggunakan metode tertentu pada database pengobatan di rumah sakit, dapat menghasilkan statistik jenis penyakit yang diderita pasien pada bulan-bulan

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

tertentu, sehingga dapat diterbitkan informasi untuk mengantisipasi penyakit tersebut pada bulan-bulan tersebut.

Secara garis besar, proses KDD dibagi menjadi tiga langkah yaitu sumber data disiapkan untuk di proses, lalu data tersebut diolah, dan akhirnya informasi yang didapat di interpretasi menjadi pengetahuan. *Knowladge Discovery* adalah proses interaksi yang berulang yang menyangkut beberapa langkah sebagai berikut (Soundararajan dkk., 2005).

1. *Learning the application domain* adalah mengetahui tujuan dari aplikasi yang akan dibuat.
2. *Target data set* adalah memilih dataset atau sample data yang akan digunakan dalam *Knowledge Discovery*.
3. *Data Cleaning and Preprocessing* adalah menghilangkan data *noise*, dan menentukan strategi dalam menangani data tidak lengkap.
4. *Data Reduction and Projection* adalah mengurangi variabel yang akan memperlambat proses dan menggunakan fitur-fitur untuk merepresentasikan data.
5. *Choosing the function of data mining* adalah memilih fungsi *data mining* berdasarkan model data yang akan digunakan seperti *clustering*, *clasification*, *regression*, dan *association*.
6. *Choosing the data mining algorithm* adalah memilih metode yang akan digunakan untuk pencarian *pattern* seperti algortima apriori, algoritma similis, dan algoritma *FP-Growth*.
7. *Data mining* adalah proses komputasi untuk mencari *pattern* dalam database, dan menentukan *pattern* yang layak dianggap sebagai informasi.

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

8. *Interpretation* adalah menginterpretasikan *pattern* yang sudah ditemukan, menghilangkan *pattern* yang berlebihan maupun yang tidak relevan, dan menerjemahkan data yang berguna agar lebih mudah dimengerti oleh user.
9. *Using the discovered Knowladge* adalah menggunakan informasi yang dihasilkan kedalam sistem, maupun mendokumentasikan untuk digunakan di kemudian hari.

Proses *Knowledge Discovery in Databases* melibatkan bebearapa proses yaitu mempersiapkan data, memilih metode dan fungsi *data mining* yang tepat, proses *data mining* untuk menemukan *pattern*, dan menginterpretasi hasil yang didapat. Salah satu proses paling penting dalam KDD ialah *data mining* (Kumar, 2011).

2.3 Data Mining

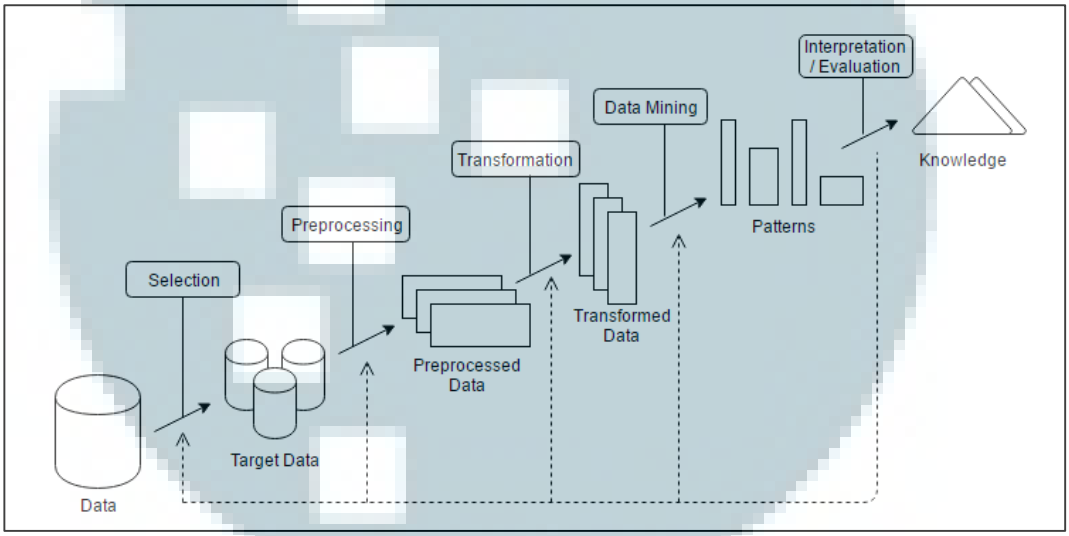
Perkembangan *data mining* dimulai sejak tahun 1980. Istilah *data mining* mulai digunakan dalam komunitas riset. Saat itu *data mining* diartikan sebagai kumpulan mekanisme dan teknik untuk mengekstrak informasi tersembunyi di dalam data (Coenen, 2004). Pada tahun 1990 *data mining* dikenal sebagai *sub-process* dalam proses yang lebih besar yang disebut *Knowledge Discovery in Databases*.

Menurut Han dkk (2012) *data mining* dapat diartikan sebagai proses komputasi untuk menemukan pola yang ada dalam kumpulan data yang menggunakan metode gabungan dari *artificial intelligence*, *machine learning*, statistik, dan sistem database. Tujuan utama dari *data mining* adalah untuk

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

menemukan informasi dari data set dan ditransformasikan ke bentuk yang lebih mudah dimengerti untuk digunakan lebih lanjut.

Data Mining merupakan bagian dari *Knowledge Discovery in Databases*, bukan sebagai teknologi yang utuh berdiri sendiri. *Data Mining* merupakan suatu bagian langkah penting dalam KDD terutama berkaitan dengan ekstrak dan perhitungan pola-pola dari data yang diteliti (Fayyad dkk., 1996). Gambar 2.1 menunjukkan langkah-langkah dalam *Knowladge Discovery in Database* (KDD).



Gambar 2.1 Langkah-langkah dalam KDD (Fayyad dkk., 1996)

Adapun penjelasan mengenai langkah-langkah dalam proses KDD sebagai berikut.

1. *Selection*

   Proses memilih dan memisahkan data berdasarkan beberapa kriteria, sebagai contoh panjualan pada bulan tertentu.

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

2. *Preprocessing*

   Mempersiapkan data dengan cara membersihkan data informasi atau *field* yang tidak dibutuhkan, yang jika dibiarkan hanya akan memperlambat proses *query*.

3. *Transformation*

   Data yang telah melalui proses *selection* dan *preprocessing* tidak begitu saja langsung digunakan, tapi ditransformasikan terlebih dahulu ke bentuk yang lebih mudah untuk diolah.

4. *Data Mining*

   Proses pencarian pola berdasarkan data atau informasi yang diteliti. Contohnya pencarian hubungan antara item yang satu dengan item yang lain dalam data penjualan.

5. *Interpretation and Evaluation*

   Dalam proses ini *pattern* atau pola yang telah diidentifikasi oleh sistem diterjemahkan ke bentuk *knowladge* yang lebih mudah dimengerti oleh pengguna aplikasi.

Data mining menemukan *pattern* dari database dengan menggunakan bermacam-macam metode tergantung jenis data yang digunakan. Metode dalam *data mining* yaitu *clustering*, *clasification*, *regression*, *association*, dan metode lainnya. *Association rule mining* adalah metode yang bertujuan untuk mencari *frequent itemset*.

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

2.4 Association Rule

Association rules adalah metode untuk menemukan relasi antara variabel dalam database. Menurut penelitian yang dilakukan oleh beberapa ahli seperti Agrawal dan Srikant pada tahun 1994, Zaki pada tahun 1999, dan Hipp dkk pada tahun 2000 menyatakan bahwa *association rules* telah dianggap sukses dalam pembentukan relasi antara variabel, namun dalam melakukan pembentukan aturan dalam database yang besar, tidak menutup kemungkinan menghasilkan aturan yang tidak masuk akal (Reader dan Chawla, 2011). Oleh karena itu pengambilan keputusan apakah rule yang dihasilkan akan digunakan dalam pembentukan strategi penjualan atau tidak tetap berada di tangan pengguna aplikasi.

Adapun beberapa komponen *association rules* yang digunakan dalam pembentukan aturan atau relasi antara variabel dalam kelompok dataset. Berikut beberapa komponen yang digunakan dalam pembentukan rule.

1. Item adalah barang yang terdapat pada setiap transaksi. Kombinasi item dalam setiap transaksi bisa saja berbeda. Item bisa saja dalam bentuk barang, jasa, artikel dalam blog, dan subyek penelitian lainnya.
2. Transaksi adalah kumpulan beberapa item yang dibeli bersamaan. Pada data penjualan, transaksi adalah itemset yang dibeli oleh pelanggan dalam satu transaksi. Misal dalam transaksi T01 mengandung item susu, telur, dan tepung, sedangkan dalam T02 mengandung item telur, gula, dan tepung.
3. Support adalah seberapa sering sebuah item muncul pada keseluruhan transaksi. Nilai support didapat dari perhitungan jumlah item muncul dalam transaksi dibagi dengan jumlah transkasi yang ada atau bisa dirumuskan dalam rumus 2.1.

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

$$Support = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}} \quad ...(2.1)$$

Adapun nilai support dari dua item yang dapat dihitung dengan rumus 2.2.

$$Support\ (\text{A} \cap \text{B}) = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}} \quad ...(2.2)$$

4. *Confidence* adalah indikasi seberapa sering kandidat aturan yang dibuat bernilai benar. Nilai confidance aturan barang A dibeli maka barang B juga akan dibeli dihitung dengan rumus 2.3.

$$Confidence\ \text{A} \rightarrow \text{B} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Jumlah Transaksi mengandung A}} \quad ...(2.3)$$

5. *Lift* adalah nilai yang menyatakan apakah suatu item berpengaruh terhadap penjualan item lainnya atau tidak. Nilai lift didapatkan dengan rumus 2.4.

$$lift(\text{A} \rightarrow \text{B}) = \frac{\text{support}(\text{A} \cup \text{B})}{\text{support}(\text{A}) \times \text{support}(\text{B})} \quad ...(2.4)$$

Jika nilai *lift* lebih besar dari 1, maka item A berpengaruh meningkatkan penjualan item B. Jika nilai *lift* lebih kecil dari 1 maka item A berpengaruh menurunkan penjualan item B. Sedangkan jika nilai *lift* bernilai 1 maka item A dan item B bersifat independen dan tidak terpengaruh satu sama lain.

Penerapan *association rule mining* menghasikan algoritma apriori yang diperkenalkan oleh Agrawal dan Srikant pada tahun 1994. Algoritma apriori merupakan salah satu pencetus berkembangnya algoritma untuk mencari *frequent itemset* (Li dkk., 2009). Salah satu pengembangan dari algoritma apriori adalah algoritma *FP-Growth*. Algoritma *FP-Growth* dianggap sebagai algoritma yang lebih unggul dalam pembuatan *fraquent pattern* dikarenakan proses nya yang lebih cepat (Garg dan Kumar, 2013).

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

2.5 Algoritma FP-Growth

Algoritma *FP-Growth* atau *Frequent Pattern Growth* adalah salah satu alternatif algortima yang digunakan untuk menentukan himpunan data yang paling sering muncul dalam sekumpulan data tanpa membuat kandidat itemset. Dengan demikian algoritma *FP-Growth* dapat mempercepat proses pembentukan struktur data. Inti dari metode *FP-Growth* adalah penggunaan struktur data *frequent-pattern tree* (*FP-Tree*) yang menampung itemset dari database (Han dkk., 2012). Setelah data ditampung dalam struktur *FP-Tree*, data akan di ekstrak dari *tree* sehingga didapatkan *frequent pattern* yang kemudian dapat diproses kembali maupun di tampilkan kepada user sebagai informasi yang lebih berguna.

2.5.1 Sejarah Singkat Algoritma FP-Growth

Sejarah algoritma *FP-Growth* berawal pada tahun 1994 oleh Rakesh Agrawal dan Ramakrishnan Srikant dalam karya ilmiahnya yang berjudul "*Fast Algorithm for Mining Association Rules in Large Database*" yang menghasilkan algoritma Apriori. Kemudian Algoritma ini menjadi acuan penelitian yang dilakukan oleh Han dkk. (2000) yang berjudul "*Mining Frequent Patterns Without Candidate Generation*" yang menghasilkan algoritma *FP-Growth*.

2.5.2 Kelebihan dan Kekurangan Algoritma FP-Growth

Berdasarkan penelitian yang dilakukan oleh Garg dan Kumar pada tahun 2013 yang berjudul *Comparing the Performance of Frequent Pattern Mining Algorithm* menyatakan bahwa algoritma *FP-Growth* merupakan algoritma yang lebih baik dibandingkan dengan algoritma apriori dikarenakan algoritma apriori yang memakan waktu saat menentukan kandidat itemset dan membaca database yang berulang-ulang, sedangkan algoritma *FP-Growth* hanya membaca database

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

satu kali saat pembuatan *FP-Tree* dan tidak menggunakan kandidat itemset dalam proses pembentukan *frequent pattern*.

Selain itu penelitian lainnya dilakukan oleh Cavique (2007) pada penelitian yang berjudul "*A Scalable Algorithm for the Market Basket Analysis*" menyatakan bahwa algoritma *FP-Growth* memiliki waktu yang lebih cepat dalam pengolahan data *market basket analysis* jika jumlah data dalam itemset lebih kecil dari lima belas item. Namun jika jumlah item dalam sebuah itemset memiliki data yang lebih dari lima belas item, maka algoritma *FP-Growth* memakan waktu yang lama dalam pembentukan *FP-Tree*. Perbandingan performa algoritma *FP-Growth* dan algoritma Similis dapat dilihat pada tabel 2.1.

Tabel 2.1 Perbandingan Performa Algoritma Similis dengan Algoritma *FP-Growth* (Cavique, 2007)

Dataset	Similis				FP-Growth	
	filter	vertex	itemset	time(sec.)	minsup	time(sec.)
Frozen-food	0.0%	159	5	33	1%	1
	0.0%	159	10	34	1%	1
	0.0%	159	15	35	0.1%	>900
	0.0%	159	20	39	-	-
retail	0.5%	222	5	309	0.1%	1
	0.5%	222	10	331	0.01%	3
	0.5%	222	15	352	0.005%	4
	0.5%	222	20	381	0.001%	>900
accidents	0.0%	468	5	102	10%	37
	0.0%	468	10	154	10%	27
	0.0%	468	15	200	5%	119
	0.0%	468	20	244	1%	>900

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

2.5.3 Tahapan Alur Algoritma FP-Growth

Algoritma *FP-Growth* ini dibagi menjadi dua tahap. Tahap pertama adalah tahap pembuatan *FP-Tree* dan tahap kedua adalah mengekstrak *itemset* yang sering muncul pada *FP-Tree*.

Pada tahap pertama algoritma *FP-Growth*, input yang digunakan adalah data transaksi yang kemudian dicari nilai support-nya dan menghilangkan data dengan support lebih kecil daripada minimum support. Process ini dilakukan agar dalam pembentukan *FP-Tree* tidak memakan waktu dalam perhitungan item dengan frekuensi yang rendah. Kemudian data akan diurutkan secara *descending* untuk mengetahui barang paling sering dibeli hingga yang paling jarang dibeli agar pembentuka node pada *FP-Tree* lebih efisien (Leers, 2011). Setelah data diurutkan dari yang paling sering hingga yang paling jarang muncul, maka dibentuk *FP-Tree* dimana *root* dari *FP-tree* adalah *null*. Setiap transaksi dalam database misalkan transaksi dengan item{I1, I3, I4} maka I1 adalah *child* dari *root*, I3 adalah *child* dari I1, dan I4 adalah *child* dari I3 (Singh dkk., 2013). Jika suatu item sudah terdaftar sebagai *child* dari *root* ataupun *parent*, maka *count* dari item tersebut akan ditambahkan. Hal yang sama akan dilakukan terhadap semua data transaksi hingga terbentuk keseluruhan *FP-Tree*.

Tahap kedua dalam algoritma *FP-Growth* adalah mengekstrak data dari *FP-Tree* untuk mencari itemset yang sering muncul. Prosedur dalam mengekstrak *FP-Tree* adalah sebagai berikut.

1. Pada setiap node yang ada pada *FP-Tree* akan dibaca nilai *count*. Setelah itu node tersebut akan dilihat *parent*-nya agar diketahui barang apa saja yang

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017

dibeli bersamaan dengan barang tersebut sehingga didapatkan itemset dan jumlah itemset yang dibeli secara bersamaan.

2. Setelah didapatkan barang yang dibeli bersamaan dengan item pada node, data tersebut akan disimpan dan kembali dilakukan pengekstrakan *tree* terhadap *node* berikutnya.
3. Hasil ekstraksi *FP-Tree* akan diurutkan berdasarkan dengan frekuensi tertinggi sehingga user dapat melihat produk yang memiliki *pattern* yang paling sering dibeli bersamaan.

Dengan demikian dihasilkan *pattern* dengan frekuensi tinggi dan informasi tersebut dapat digunakan oleh pengguna aplikasi untuk mengatur strategi penjualan seperti membentuk *package bundling* berdasarkan *frequent pattern* sehingga diharapkan dapat meningkatkan penjualan *cross-selling*.

Rancang bangun...,Alvin Kurniawan,FTI UMN, 2017