TINJAUAN PUSTAKA

1.1. Penelitian terkait

Tabel x.x merupakan penelitian-penelitian terdahulu yang terkait dengan object pada penelitian ini.

Tabel x.x Penelitian Terkait

Peneliti	**Judul**	**Object**	**Metode**	**Tujuan**
Christioko, B.V. Nugroho, A. Khoirudin (2020)	Algoritma Hash Based untuk Menemukan Pola Asosiatif Data Tracer Study	Data tracer study	Algoritma Hash Based	Menghasilkan association rules atau aturan asosiasi
Zahrotun, Lisna Setiadi, Tedy Haryadi, Taufik Mufti (2018)	Aplikasi Data Mining untuk Mencari Pola Asosiasi Tracer Study Menggunakan Algoritma FOLDARM	Data tracer study	Algoritma FOLDARM	Menghasilkan association rules atau aturan asosiasi
Nirad, Dwi Welly Sukma Surendro, Kridanto (2018)	Analisis Data Tracer Study Dengan Mengidentifikasi Outlier	Mahasiswa drop out dan memiliki penghasilan tinggi pada masa berkarir.	Association rule mining dan Outlier	validitas derajat mahasiswa outlier serta rekomendasi keputusan dalam menangani mahasiswa yang teridentifikasi sebagai outlier
Abdulloh, Ferian Fauzi Kusnawi (2017)	IMPLEMENTASI DATA MINING UNTUK MENEMUKAN POLA ASOSIATIF DATA TRACER STUDY	Data tracer study	Algoritma Apriori	informasi pola hubungan antar atribut pada tracer study
Fana Wiza (2016)	Pemodelan Pola Hubungan Kemampuan Lulusan Universitas Lancang Kuning Dengan Kebutuhan Dunia Usaha dan Industri	Data tracer study	Algoritma Apriori	informasi pola hubungan antar atribut pada tracer study

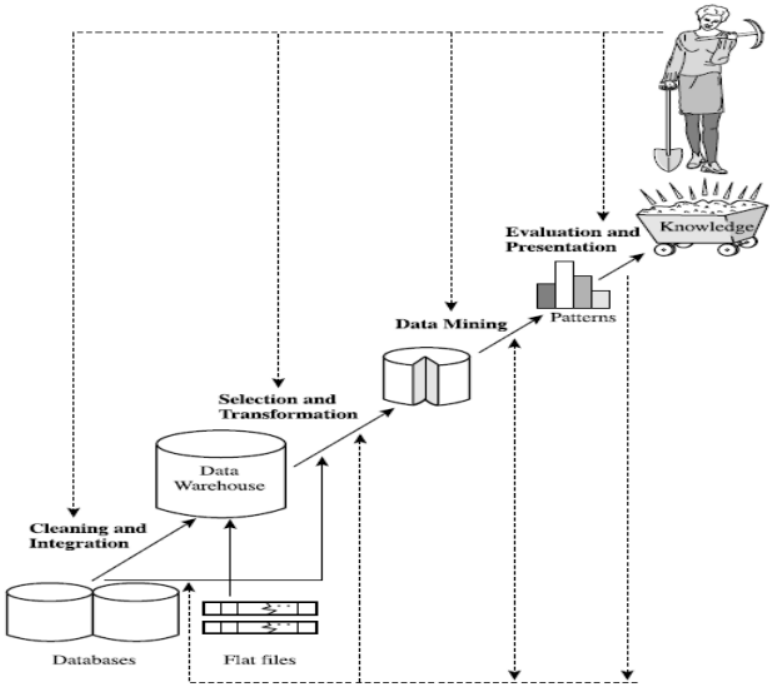
1.2. Landasan Teori

1.2.1. Data Mining

Dalam penggunaan database banyak proses dan transaksi data yang tersimpan dalam penyimpanan. Pengumpulan data yang melimpah secara terus menerus akan mengakibatkan penumpukan pada data storage. Maka dari itu, diperlukan suatu teknik yang dapat mengotomasi dan mentransformasikan data-data tersebut untuk menghasilkan informasi dan pengetahuan yang berguna. Data mining adalah serangkaian proses untuk mengggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual.

Kata mining berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. Data mining merupakan proses pencarian pola dan relasi- relasi yang tersembunyi dalam sejumlah data yang besar dengan tujuan untuk melakukan klasifikasi, estimasi, prediksi, association rule, clustering, deskipsi dan visualisasi (Han, dkk, 2012). Data Mining adalah suatu proses yang menggunakan teknologi pengenalan pola serta teknik statistik dan matematika untuk menemukan korelasi baru yang mempunyai arti, pola, dan tren dengan cara memilah dan menyaring data yang tersimpan dalam data storage (Agarwal, 2013).

Kata lain dari data mining adalah Knowledge Discovery from Database (KDD). Gambar 2.1 dibawah ini menunjukkan proses KDD secara umum yang terdiri dari langkah- langkah (Han dkk, 2012), yaitu:



Gambar 2.1 Tahap dalam proses Discovery Knowledge (Han, dkk, 2012)

1. Pembersihan data (data cleaning), proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan.
2. Melakukan integrasi data (data integration), penggabungan data dari berbagai basis data ke dalam satu basis data baru.
3. Pemilihan data (data selection), pemilihan data relevan yang didapat dari basis data.
4. Transformasi data (data transformation), data diubah ke dalam format yang sesuai untuk diproses dalam data mining.
5. Data mining, suatu proses dimana metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.
6. Evaluasi pola (pattern recognition), untuk mengidentifikasi pola- pola menarik untuk direpresentasikan ke dalam knowledge based.

7.
8. Representasi pengetahuan (knowledge presentation), visualisasi dan penyajian pengetahuan mengenai teknik yang digunakan untuk memperoleh pengetahuan yang diperoleh oleh user

1.2.2. *Association Rules*

Analisis asosiasi atau association rules adalah suatu proses untuk menemukan pola aturan assosiatif yang memenuhi syarat minimum untuk support (minsup) dan syarat minimum untuk confidence (mincof) pada database. Association rule mining adalah suatu prosedur untuk mencari hubungan antar item dalam suatu dataset yang ditentukan. Association rule mining meliputi dua tahap (Han dkk, 2012). Tahapan tersebut adalah:

1. Mencari kombinasi yang paling sering terjadi dari suatu itemset yang memiliki support diatas minimal support. Hal ini disebut dengan frequent itemset.
2. Mendefinisikan association rules dari frequent itemset yang telah dibuat berdasarkan aturan minimal support dan minimal confidence.

Terdapat dua ukuran yang digunakan dalam menentukan suatu association rule, yaitu support dan confidence. Kedua ukuran ini nantinya berguna dalam menentukan interesting association rules, yaitu untuk dibandingkan dengan batasan yang telah ditentukan. Batasan tersebut terdiri dari minsup dan mincof

1. Minimal Support

Minimal support adalah suatu ukuran atau nilai yang harus dipenuhi sebagai batasan besar frekuensi kejadian (support count) dari seluruh nilai dominasi suatu item atau itemset (support) dalam keseluruhan transaksi (Han, dkk, 2012). Nilai support sebuah item (misal: X) diperoleh dengan rumus :

$$Support\ (X) = \frac{\text{Jumlah Transaksi Mengandung X}}{\text{Total Transaksi}} \qquad [2.1]$$

Sedangkan nilai support dari itemset (misal: X,Y) diperoleh dari rumus :

$$Support\ (X,Y) = \frac{\text{Jumlah Transaksi Mengandung X dan Y}}{\text{Total Transaksi}} \qquad [2.2]$$

2. Minimal Confidence

Nilai minimal confidence merupakan parameter yang mendefinisikan minimum level suatu nilai hubungan antar item (confidence) yang harus dipenuhi agar menemukan aturan yang berkualitas (Han, dkk, 2012). Menghitung nilai confidence assosiatif X darisupport pola frequent itemset X dan Y dengan menggunakan rumus :

$$Confidence\ (Y \backslash X) = \frac{\text{Jumlah Transaksi Mengandung X dan Y}}{\text{Jumlah Transaksi X}} \qquad [2.3]$$

1.2.3. Algoritma Hash- based

Untuk menemukan aturan asosiasi dari transaksi database diperlukan pembentukan kandidat (k+1)- itemset yang berasal dari large k- itemset. Pembentukan tersebut dihitung berdasarkan kejadian dari kandidat (k+1)- itemset melalui penelusuran seluruh record dalam database. Semakin banyak record database maka semakin besar pula jumlah kandidat itemset yang terbentuk sehingga waktu yang dibutuhkan untuk menghasilkan frequent itemset semakin lama. Untuk mempersingkat waktu dan mengefisienkan kinerja penambangan data maka dibutuhkan sebuah teknik atau algoritma yang bisa mengurangi ukuran kandidat k- itemset atau Ck untuk k>1. Han dkk, (2012) menjelaskan beberapa variasi untuk meningkatkan efisiensi dari algoritma Apriori. Salah satu dari variasi tersebut adalah Hash- based technique. Algoritma Hash Based dapat digunakan untuk

mengurangi jumlah kandidat k- itemset. Efisiensi frequent itemset terjadi pada saat pembangkitan kandidat itemset terutama pada frequent 2- itemset. Hal ini dapat meningkatkan performa dari data mining.

Algoritma Hash Based menggunakan teknik hashing untuk menyaring keluar itemset yang tidak penting untuk pembangkitan itemset selanjutnya. Ketika support count untuk kandidat k- itemset dihitung dengan menelusuri database, algoritma hash- based mengumpulkan informasi mengenai (k+1)- itemset dengan cara seluruh kemungkinan (k+1)- itemset dihash kedalam hash table denganmenggunakan fungsi hash (yang menggunakan sebuah bilangan prima untuk operasi modulo). Setiap bucket pada hash table berisi angka berapa kali itemset telah dihash kedalam bucket tersebut. Berdasarkan hash table tersebut akan dibangun bit vector yang dimana bit vector bernilai 1 jika angka pada bucket yang bersangkutan lebih besar atau sama dengan minimum support. Pada bagian pembangkitan kandidat, setelah menghitung Ck =Lk-1 *Lk-1 , setiap k- itemset diperiksa apakah itemset tersebut di- hash ke bucket yang memiliki bit vector sama dengan satu. Bila tidak maka itemset tersebut tidak akan digunakan. Penggunaan hash table ini mengurangi jumlah kandidat k- itemset, sehingga mampu mengurangi nilai komputasi dari pembangkitan itemset pada setiap iterasi.

Untuk membangkitkan frequent itemset, algoritma hash- based terbagi menjadi tiga bagian utama yaitu

1. Algoritma hash- based akan menghasilkan C_1 (kandidat 1- itemset) dan L_1 (large 1 - itemset) dari database.Untuk kandidat 1- itemset,seluruh transaksi ditelusuri untuk menghitung support count dari itemset ini. Pada tahap ini hash tree untuk C_1 dibangun dengan tujuan mengefisienkan perhitungan support count. Algoritma Hash Based memeriksa apakah setiap item sudah ada dalam hash tree. Jika sudah ada,maka jumlah dari item ini ditambah satu. Tetapi jika belum ada maka item ini dimasukan dengan jumlah sama dengan satu ke dalam hash tree. Pada bagian ini juga algoritma akan membangun hash table (dengan fungsi hash) untuk 2 itemset yang akan berguna unutk mengurangi banyaknya kandidat 2- itemseti, C_2.
2. Kumpulan kandidat itemset C_k dibangkitkan berdasarkan hash table yang telah dibuat pada iterasi sebelumnya. Lalu ditentukan frequent itemset L_k dan mengurangi ukuran database untuk pembangkitan itemset selanjutnya. Selain itu hash table untuk kandidat (k+1)- itemset juga akan gian dari algoritma ini terbagi lagi menjadi dua fase. Fase pertama untuk membangjitkan kandidat k-itemset C_k berdasarkan pada Hash Based H_k , proses ini terjadi pada prosedur gen_candidate. Algoritma Hash Based membangkitkan k- itemset dengan L_k-1, tetapi yang unik disini digunakan bit vector untuk menguji validitas dari setiap k- itemset. Untuk seluruh itemset dari L_k- 1* L_k-1 , algoritma Hash Based hanya menambahkan k- itemset yang melewati penyaring untuk dimasukan kedalam C_k yang merupakan hash tree. Fase kedua dari bagian kedua algoritma Hash Based akan menghitung support dapa kandidat itemset dan mengurangi ukuran dari setiap transaksi transaksi, proses ini terjadi pada prosedur count_support.Setelah itu akan dibangkitkan hash table untuk (k+1)- itemset dan dilakukan lagipemangkasan data transaksi pada prosedur make_hash.
3. Seperti pada bagian kedua namun tidak menggunakan hash table sehingga mirip dengan algoritma apriori. Bagian kedua dilakukan selama nilai hash bucket lebih besar dari minimum support. Setelah batasan ini terlewati, algoritma Hash Based diganti dengan algoritma apriori karena tidak lebih efisien dibandingkan algoritma apriori.