Journal of Informatics and Telecommunication Engineering

Available online http://ojs.uma.ac.id/index.php/jite

Analisa Algoritma Data Mining Eclat Dan Hui Miner

Analysis of Data Mining Eclat and Hui Miner Algorithms

Juanda Hakim Lubis*
Program Studi Teknik Informatika, Fakultas Teknik
Universitas Medan Area, Indonesia

*Corresponding author: E-mail : juandahakim@gmail.com

Abstrak

Frequent pattern mining memainkan peran penting di dalam data mining. Salah satu metode yang digunakan adalah metode asosisi. Metode asosiasi digunakan untuk mencari dan menganalisa data transaksi penjualan yang terjadi. Hal ini dapat dilakukan dengan memeriksa perilaku pelanggan terkait dengan produk - produk yang dibeli. Dengan menggunakan aturan asosiasi, kita dapat mengetahui seberapa sering item yang dibeli bersama-sama dalam suatu transaksi. Salah satu algoritma yang digunakan adalah Eclat. Eclat pada dasarnya adalah pencarian algoritma depth-first menggunakan persimpangan yang ditetapkan. Kelebihan dari Eclat adalah proses dan performa penghitungan support dari semua itemsets dilakukan dengan lebih efisien dibandingkan dengan algoritma HUI-miner apriori. Akan tetapi dalam penelitian ini diperoleh hasil algoritma HUI-miner lebih efektif dan lebih stabil dari segi waktu jika di bandingkan dengan eclat.

Kata Kunci : *Algoritma Eclat; Algoritma Hui-Miner; Datamining; Frequent pattern mining*

Abstract

Frequent pattern mining memainkan peran penting di dalam data mining. Salah satu metode yang digunakan adalah metode asosiasi. Metode asosiasi digunakan untuk mencari dan menganalisa data transaksi penjualan yang terjadi. Hal ini dapat dilakukan dengan memeriksa perilaku pelanggan terkait dengan produk - produk yang dibeli. Dengan menggunakan aturan asosiasi, kita dapat mengetahui seberapa sering item yang dibeli bersama-sama dalam suatu transaksi. Salah satu algoritma yang digunakan adalah Eclat. Eclat pada dasarnya adalah pencarian algoritma depth-first menggunakan persimpangan yang ditetapkan. Kelebihan dari Eclat adalah proses dan performa penghitungan support dari semua itemsets dilakukan dengan lebih efisien dibandingkan dengan algoritma HUI-miner apriori. Akan tetapi dalam penelitian ini diperoleh hasil algoritma HUI-miner lebih efektif dan lebih stabil dari segi waktu jika di bandingkan dengan eclat.

Keywords : *Algoritma Eclat; Algoritma Hui-Miner; Datamining; Frequent pattern mining*

How to Cite: Hakim, J. 2017, Analisa Algoritma Data Mining Eclat Dan Hui Miner, *Journal of Informatics and Telecommunication Engineering, 1(1) :8-13.*

PENDAHULUAN

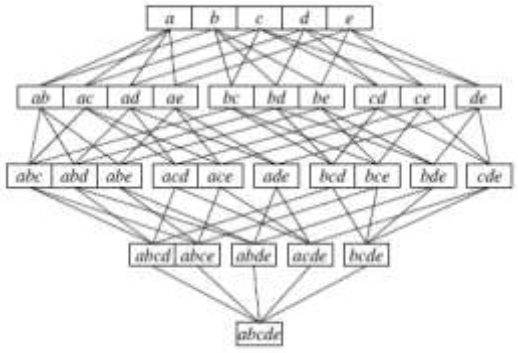
Saat ini, pertumbuhan jumlah toko yang menyediakan perlengkapan suku cadang dan aksesoris sepeda motor semakin banyak. Selain itu perbedaan harga produk – produk yang ditawarkan juga tidak jauh berbeda antara satu toko dengan toko yang lain. Supaya suatu toko dapat memiliki keunggulan dengan toko lainnya, salah satu cara yang dapat ditempuh adalah dengan mengetahui pola beli pelangan dengan menganalisa data transaksi penjualan. Dengan mengetahui pola beli pelanggan, diharapkan toko dapat mengetahui jenis barang yang sering dibeli bersamaan sehingga dapat menambah keuntungan penjualan.

Frequent pattern mining memainkan peran penting di dalam data mining. Frequent pattern mining memiliki tujuan untuk menemukan pola tertentu dari database. Salah satu metode yang digunakan adalah metode asosisi. Metode asosiasi digunakan untuk mencari dan menganalisa data transaksi penjualan yang terjadi. Hal ini dapat dilakukan dengan memeriksa perilaku pelanggan terkait dengan produk - produk yang dibeli. Dengan menggunakan aturan asosiasi, kita dapat mengetahui seberapa sering item yang dibeli bersama-sama dalam suatu transaksi. Salah satu algoritma yang digunakan adalah Eclat. Eclat pada dasarnya adalah pencarian algoritma depth-first menggunakan persimpangan yang ditetapkan. Eclat menggunakan basis data dengan tata letak vertikal. Setiap item disimpan bersama dengan sampulnya (juga disebut tidlist) dan menggunakan pendekatan berdasarkan persimpangan untuk menghitung dukungan dari suatu itemset. Dengan cara ini, dukungan dari itemset X dapat dengan mudah dihitung dengan hanya memotong penutup dari dua himpunan bagian. Kelebihan dari Eclat adalah proses dan performa penghitungan support dari semua itemsets dilakukan dengan lebih efisien dibandingkan dengan algoritma HUI-miner apriori.

Association rule mining adalah salah satu metode untuk mencari relasi yang menarik antara variabel pada sebuah basis data yang besar. Konsep ini diperkenalkan oleh Agrawal et al1 dengan menggunakan kasus transaksi pada supermarket yang disimpan pada sistem point of sales (POS) untuk menemukan barang yang dibeli bersama oleh konsumen. Metode ini dinamakan market basket analysis.

Market basket analysis adalah sebuah metode untuk mencari set item yang sering dalam satu set transaksi. Tujuan dari market basket analysis adalah untuk menemukan perilaku atau pola belanja pelanggan supermarket, perusahaan mail-order, toko online, dll. Secara khusus, market basket analysis akan mencoba untuk mengidentifikasi set produk yang sering dibeli bersama-sama.

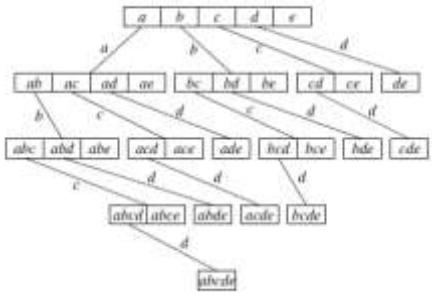
Masalah utama di dalam menemukan itemset, misalnya yang terkandung di dalan transaksi adalah terdapat banyak sekali kemungkinan itemset yang muncul, yang membuat pendekatan naif menjadi tidak layak karena waktu eksekusi yang lama (Goethals, 2013)(2). Namun, ada pendekatan yang lebih canggih dengan dua algoritma dikenal dengan nama Apriori dan Eclat yang paling populer. Keduanya mengandalkan pencarian top-down dalam kisi subset dari item. Contohnya pada gambar 1 yang menunjukkan relasi antara item set yang berbeda.



Gambar 1. Subset bagian untuk lima item

Untuk struktur pencarian, kedua algoritma mengatur kisi bagian sebagai pohon prefix, misalnya pada gambar 2 memiliki lima item. Dalam pohon set item yang digabungkan dalam sebuah node yang memiliki prefix yang sama berkaitan dengan beberapa perintah yang acak,

namun memiliki urutan tetap di dalam item (dalam contoh item lima, urutan ini hanya a, b, c, d, e). Dengan struktur ini, item set yang terkandung dalam simpul dari pohon dapat dibangun dengan mudah dengan cara berikut: Ambil semua item yang bagian tepinya mengarah ke node memiliki label dan tambahkan item yang berhasil dengan urutan yang tetap dari item. Perhatikan bahwa dengan cara ini hanya memerlukan satu item untuk membedakan antara item set yang diwakili dalam satu node yang relevan untuk pelaksanaan algoritma



Gambar 2. Pohon prefix untuk lima item

Algoritma Eclat (Equivalence Class Transformation) adalah sebuah program yang digunakan untuk menemukan set item yang sering (Zaki et al. 1997)[4], algoritma ini menggunakan yang melakukan pencarian depth first pada kisi bagian dan menentukan dukungan set item dengan memotongkan daftar transaksi. Versi saat ini dari program ini hanya dapat menemukan set item yang sering. Algoritma ini tidak mendukung pengelompokan item / clustering (Zaki et al, 1997.)[4], tetapi algoritma ini dapat mendukung diffset (Zaki dan Gouda 2003)[5] dan beberapa varian algoritma lainnya.

Algoritma Eclat melintasi pohon prefix di urutan pertama. Algoritma ini memperluas item set barang hingga mencapai batas antara item set yang sering dan jarang terjadi, kemudian kembali ke awal untuk mengerjakan awalan berikutnya (dalam urutan leksikografis berkaitan dengan urutan tetap dari item)[6]. Eclat menentukan support dari item set yang ditetapkan dengan membangun daftar pengenal transaksi yang mengandung item set. Caranya dengan memotong dua daftar pengenal transaksi di dalam dua set item yang berbeda hanya dengan satu item dan bersama-sama membentuk item yang ditetapkan saat diproses.

Sebuah cara mudah untuk merepresentasikan transaksi untuk algoritma Eclat adalah dengan menggunakan matriks bit, di mana setiap baris sesuai dengan item masing-masing kolom untuk transaksi (atau sebaliknya). Bit adalah set di dalam matriks ini jika item yang sesuai dengan baris terkandung di dalam transaksi yang sesuai dengan kolom, selain itu akan dihapus.

Pada dasarnya ada dua cara untuk merepresentasikan matriks bit: Baik sebagai matriks bit yang benar, dengan satu bit memori untuk setiap item dan transaksi, atau menggunakan daftar untuk setiap baris di dalam kolom bit yang ditetapkan. (cara kedua ini setara dengan menggunakan daftar pengenal transaksi untuk setiap item.). Cara merepresentasikan bit matrix ini tergantung pada kepadatan dataset. Pada mesin 32 bit, representasi bit matrix dengan cara pertama akan menggunakan memori dengan lebih efisien jika rasio dari bit set ke bit bersih lebih besar dari 1:31. Namun, penggunaan rasio ini tidak dianjurkan untuk untuk memutuskan antara matrix bit yang benar dan jarang, karena dalam proses pencarian, karena saat persimpangan dilakukan, jumlah bit set akan berkurang. Oleh karena itu representasi jarang harus digunakan bahkan jika rasio dari bit untuk bit bersih lebih besar dari 1:31 (Bogelt, C.2003)[3].

Pseudocode algoritma Eclat dapat dilihat pada gambar dibawah ini

```
Input : database D , minimum support , a set of atim of a sublattice S
Output : Frequent itemsets F
Procedure Eclat ( S )
For all atoms A_i ∈ S
  T_i = θ ;
  For all atoms A_j ∈ S , with j > i do
    R = A_i ∪ A_j ;
    L(R) = L(A_i) ∩ L(A_j)
    If support(R) ≥ minsup then
      T_i = T_i ∪ {R} ;
      F_{i+1} = F_{i+1} ∪ {R} ;
    end
  end
end
```

Gambar 3. Pseudocode Eclat

(Xu, G.2011)[7]. Web mining and social networking

Algoritma akan menghasilkan itemset sering dengan memotong tidlist dari semua pasangan yang berbeda dari atom dan mengevaluasi dukungan dari kandidat berdasarkan hasil tidlist (baris 5-6). program akan memanggul prosedur rekursif dengan itemset sering yang ditemukan pada tingkat saat ini (baris 11). Proses ini berakhir ketika semua itemset sering telah dilalui. Untuk menyimpan penggunaan memori, setelah semua itemset sering untuk tingkat berikutnya telah dihasilkan, itemset pada tingkat saat ini dapat dihapus.

Dalam data mining dan association rule learning, lift adalah pengukuran kinerja model penargetan (aturan asosiasi) untuk memprediksi atau mengklasifikasikan kasus yang memiliki peningkatan respon, diukur terhadap target model pilihan yang acak. Sebuah model penargetan melakukan pekerjaan yang baik jika respon dalam target jauh lebih baik daripada rata-rata untuk populasi secara keseluruhan. Kurfa lift juga dapat dianggap sebagai variasi dari kurva penerima operasi karateristik, yang dikenal sebagai kurfa lorenz (Tufféry. 2011)[8].

Aturan lift didefinisikan sebagai

$$lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)}$$

dengan mengamati rasio dukungan untuk mengetahui apakah X dan Y adalah independen.

Misalnya, populasi memiliki tingkat respon rata-rata 5%, tetapi model tertentu telah mengidentifikasi segmen dengan tingkat respon 20%. Kemudian segmen yang akan memiliki lift 4.0 (20% / 5%). Misalnya, terdapat sebuah data yang telah digali

Tabel 1. Contoh data

Antecedent	Consequent
A	0
A	0
A	1
A	0
B	1
B	0
B	1

Dimana antecedent adalah infomasi yang dapat kita kontrol, sedangkan consequent adalah variabel yang akan dicoba untuk diprediksi.

Sebagian besar dari algotitma data mining akan mempertimbangkan aturan ini:

- Aturan 1 : A adalah 0
- Aturan 2 : B adalah 1

Support dari aturan 1 adalah 3/7 yang terdiri dari antecedent A dan consequent 0. Support dari aturan 2 adalah 2/7 yang terdiri dari antecedent B dan consequent 1.

Support dapat ditulis sebagai berikut :

$$supp(A \Rightarrow 0) = P(A \wedge 0) = P(A)P(0|A) = P(0)P(A|0)$$
$$supp(B \Rightarrow 1) = P(B \wedge 1) = P(B)P(1|B) = P(1)P(B|1)$$

Confidence untuk aturan 1 adalah 3/4 karena tiga dari 4 data di antecedent A memiliki consequent 0. Sedangkan

confidence di aturan 2 adalah 2/3 karena dua dari tiga data di antecedent B memiliki consequent 1. Confidence dapat ditulis sebagai berikut :

$$conf(A=0)=P(0|A)$$
$$supp(B=1)=P(1|B)$$

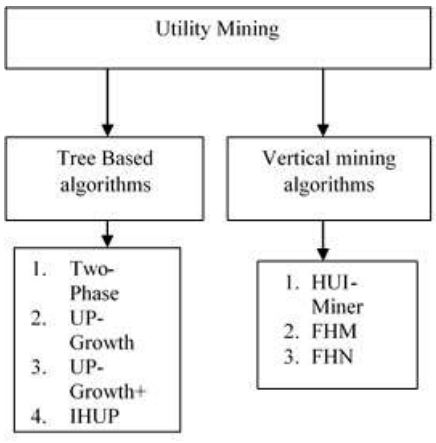
Lift dapat dicari dengan membagi support dengan probabilitas dari anteseden. Sehingga:

- lift dari aturan 1 adalah 3/4 : 4/7 = 1.3125
- lift dari aturan 2 adalah 2/3 : 3/7 = 1.56

$$lift(A=0)=\frac{P(0|A)}{P(0)}=\frac{P(A\wedge 0)}{P(A)P(0)}$$
$$lift(B=1)=\frac{P(1|B)}{P(1)}=\frac{P(B\wedge 1)}{P(B)P(1)}$$

Jika sebuah aturan memiliki lift 1, dapat dikatakan bahwa probabilitas kemungkinan munculnya antecedent dan consequent adalah independent. Ketika dua event independent, maka tidak ada aturan yang dapat disimpulkan. Jika lift bernilai positif, maka probabilitas kemungkinan munculnya antecedent dan consequent adalah dependent, sehingga dapat dibuat aturan untuk memprediksi munculnya data [9].

Algoritma dalam high-utility itemset mining dapat diklasifikasikan kedalam 2 perbedaan paradigm yaitu algoritma tree-based dan algoritma vertical mining



Algoritma vertical mining menggunakan inverted-list seperti struktur data untuk pekerjaannya. Algoritma ini pertama-tama menghasilkan semua high-utility itemset tunggal dan kemudian melanjutkan ke generasi kedua, ketiga dan seterusnya. Namun, solusi dasar pendekatan vertical mining sederhana dan telah terbukti berkinerja lebih baik dibandingkan dengan pendekatan tree-based algoritma. Namun, biaya join operation umumnya lebih tinggi untuk small-size itemset dibandingkan dengan biaya join operation untuk large-size itemset.

Pada algoritma HUI-Miner, Algoritma ini menemukan high-utility itemset tanpa generasi kandidat. Ini menciptakan sebuah struktur vertikal bernama Utility-List (UL) untuk setiap item dan kemudian menemukan high-utility item darinya. Algoritma 2 menunjukkan pseudo-code dari HUI-Miner. Di sini, daftar Utility berisi itemset, utilitas transaksi dan utilitas item; Utilitas internal adalah kuantitas yang terkait dengan iutils (utilitas internal) dalam transaksi T.

Algorithm 2: *HUI-Miner* Algorithm

Input: $P.UL$, the utility-list of itemset P, initially empty;
ULs, the set of utility-lists of all P's 1-extensions;
$minutil$, the minimum utility threshold.
Output: all the high utility itemsets with P as prefix.

```
1  foreach utility-list X in ULs do
2      if SUM(X.iutils) ≥ minutil then
3          output the extension associated with X;
4      end
5      if SUM(X.iutils)+SUM(X.rutils) ≥ minutil then
6          exULs = NULL;
7          foreach utility-list Y after X in ULs do
8              exULs = exULs+Construct(P.UL, X, Y);
9          end
10         HUI-Miner(X, exULs, minutil);
11     end
12 end
```

Gambar 4. Pseudocode HUI-Milner

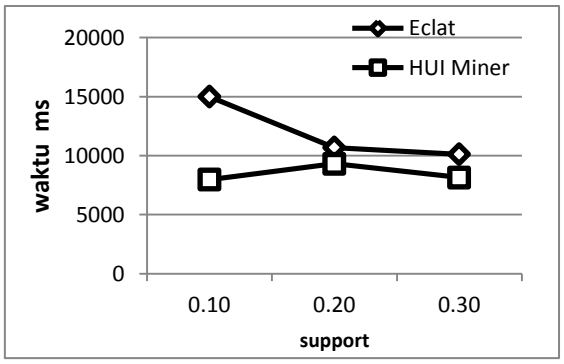
HASIL DAN PEMBAHASAN

Ujicoba data menggunakan dataset *kosarak.dat* yang memiliki baris transaksi 990.002 dan 41.270 items.

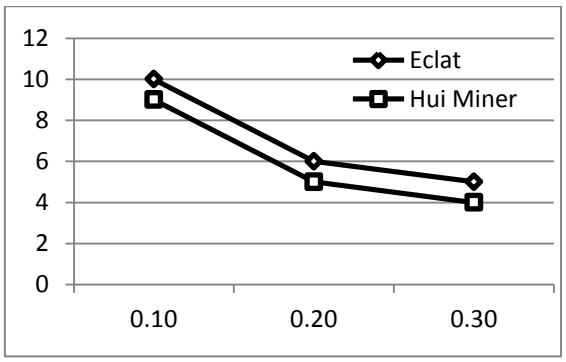
Tabel 2. Hasil Analisa waktu dan jumlah Frequent Itemsets

Minsup	Eclat		HUI Miner	
	ms	FI	ms	FI
0.10	15001	10	7969	9
0.20	10687	6	9312	5
0.30	10110	5	8152	4

Dari hasil tabel tersebut dapat disajikan dalam bentuk grafik seperti berikut :



Gambar 5. Hasil waktu proses pada algoritma



Gambar 6. hasil frequent itemset yang di dapatkan pada datset kosarak.dat dengan algoritma.

SIMPULAN

Kesimpulan yang bisa diambil dari hasil uji data adalah Hui miner lebih efektif dari segi waktu jika di bandingkan dengan eclat. Frequent yang ditemukan hanya berbeda satu buat itemset saja di setiap perubahan nilai minimum support. Hui miner lebih stabil dalam masalah penggunaan waktu.

DAFTAR PUSTAKA

Agrawal, R., Imielinski, T., dan Swami, A.N. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD International Conference on Management of Data, ACM Press.

Bogelt, C. (2003). Efficient Implementations of Apriori and Eclat. School of Computer Science, Otto-von-Guericke-University of Magdeburg

Goethals, B. (2003). Survey on Frequent Pattern Mining. University of Helsinki

Jacek, B. (2003). Handbook on Data Management in Information System.

Zaki, M. dan Gouda, K. (2003). Fast Vertical Mining Using Diffsets Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. New York, NY, USA :ACM Press.

Zaki, M., Parthasarathy, S., Ogihara, M., dan Li, W. (1997). New Algorithms for Fast Discovery of Association Rules. Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining. Menlo Park, CA, USA: AAAI Press .