PERBANDINGAN KINERJA PERANGKAT LUNAK DATA MINING UNTUK PENCARIAN POLA ASOSIASI DENGAN METODE GRAF ASOSIASI DAN METODE DIMENSI FRAKTAL

Arif Djunaidy dan Rully Soelaiman

Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl.Raya ITS – Sukolilo, Surabaya 60111, Indonesia
Telp. (031) 5939214, Fax. (031) 5939363
Email: arif@its-sby.edu

ABSTRAK

Makalah ini membahas perbandingan kinerja dari dua perangkat lunak data mining untuk menemukan pola asosiasi dari suatu basis data. Perangkat lunak yang pertama didasarkan pada metode yang berbasis pada graf asosiasi, sedang perangkat lunak yang kedua didasarkan pada penerapan metode dimensi fraktal.

Untuk memperoleh satu set pola asosiasi, pengguna dari kedua perangkat lunak harus menspesifikasikan item-item dalam bagian *antecedent* dan *consequent* pada sistem antar-muka yang disediakan oleh masing-masing perangkat lunak. Pada perangkat lunak yang didasarkan pada graf asosiasi, kualitas pola asosiasi yang dicari hanya didasarkan pada parameter *minimum support* dan *minimum confidence*. Sedang pada perangkat lunak yang didasarkan pada dimensi fraktal, selain kedua parameter tersebut, dua parameter tambahan dilibatkan dalam mengukur kualitas pola asosiasi yang dihasilkan, yaitu *window support* dan *nilai ambang batas fraktal*.

Hasil kajian perbandingan terhadap kinerja dari kedua perangkat lunak secara umum dapat disimpulkan bahwa metode dimensi fraktal dapat menghasilkan jumlah asosiasi yang jauh lebih banyak dibandingkan metode yang didasarkan pada graf asosiasi. Selain itu, waktu komputasi yang diperlukan oleh metode dimensi fraktal jauh lebih kecil dibandingkan dengan metode graf asosiasi untuk spesifikasi pola asosiasi yang sama.

Kata kunci: data mining, graf asosiasi, dimensi fraktal, pola asosiasi, minimum support, minimum confidence window support.

1. PENDAHULUAN

Pada saat ini data mining yang juga dikenal sebagai satu teknik yang dapat digunakan untuk memperoleh pola-pola tertentu yang sulit untuk ditemukan dari suatu basis data yang besar telah menjadi satu topik yang menarik dan menjanjikan dalam bidang rekayasa data. Hal ini didukung oleh kenyataan bahwa pada saat ini data mining telah mulai digunakan secara luas dalam industri, terutama sekali dalam bisnis ritel, untuk melakukan analisis sekumpulan data terdahulu dan mengekstrak beberapa pola yang dapat ditemukan secara implisit dari suatu basis data. Dalam kontek ini, oleh karena data yang dianalisis biasanya melibatkan basis data yang besar, maka teknik data mining yang digunakan haruslah melibatkan algoritma-algoritma yang dapat memberikan kinerja yang efisien dan realisitik [5, 10, 11].

Satu kemajuan yang berarti telah diperoleh dalam dasawarsa terakhir, dimana sejumlah teknik-teknik baru yang ditemukan digunakan untuk kebutuhan klasifikasi data, analisis klasterisasi dan eksplorasi pola-pola sekuensial dan asosiasi. Khusus untuk aplikasinya dalam proses pencarian pola-pola asosiasi, berbagai metode mulai dari yang didasarkan pada pendekatan statistik hingga yang secara khusus dikembangkan untuk menyelesaikan persoalan-persoalan yang melibatkan basis data yang sangat besar, telah berhasil dikembangkan [1, 8-11].

Data mining dapat didefinisikan sebagai suatu metode pencarian informasi yang tersebunyi dan merupakan bagian yang penting dalam suatu basis data berukuran besar yang sulit untuk diperoleh dengan hanya menggunakan *query* basis data biasa atau analisis statistik biasa [5]. Dalam penerapannya, data mining dapat diimplementasikan dengan menggunakan berbagai teknik, seperti jaringan syaraf, kecerdasan buatan, pembelajaran mesin otomatis, statistika, sistem berbasis pengetahuan, dan lain sebagainya. Bahkan teknik *fraktal* yang sebelumnya hanya banyak digunakan dalam lingkup pemrosesan citra, dapat juga diaplikasikan untuk mengembangkan perangkat lunak data mining [1].

Secara umum, data mining dapat diklassifikasikan dalam dua kelompok utama, yaitu *predictive data mining* dan *knowledge discovery*. Kelompok yang pertama merupakan salah satu model yang membangun data dengan menarik kesimpulan dari

Makalah diterima [16 Oktober 2001]. Revisi akhir [25 November 2001]

sekumpulan data yang besar yang kemudian berupaya untuk memperkirakan pola dari data tersebut. Sedang kelompok yang kedua menggambarkan pola data dalam bentuk ringkasan dan menampilkan sifat dari pola data yang diinginkan [5]. Dalam prakteknya, perhitungan manual sebenarnya masih dapat digunakan untuk menentukan pola data untuk data dengan jumlah record dan attribut yang kecil. Namun untuk ukuran data yang sangat besar akan memerlukan waktu yang sangat lama bahkan tidak mungkin untuk dilakukan secara manual. Dalam kontek inilah data mining dengan segala kemampuannya diharapkan mampu untuk mengatasi masalah proses pencarian pola dan masalah lainnya yang berkaitan dengan proses pencarian informasi dari suatu basis data yang besar.

Khusus untuk pencarian pola-pola asosiasi, dalam penelitian sebelumnya telah berhasil dikembangkan satu perangkat lunak data mining yang dapat digunakan untuk menemukan kaidah asosiasi dari suatu basis data yang besar sesuai dengan spesifikasi yang ditentukan oleh pengguna [2]. Namun, satu kelemahan dari perangkat lunak tersebut adalah ketidakmampuannya untuk menemukan jumlah kaidah asosiasi yang lebih besar yang sebenarnya masih dapat ditemukan dalam basis data yang digunakan. Kelemahan ini terutama terkait dengan simplifikasi yang digunakan dalam algoritma heuristik yang digunakan di dalamnya. Untuk itulah maka dalam penelitian berikutnya dicoba untuk menggunakan metode dimensi fraktal [4], yang dengan kemampuannya untuk melakukan improvisasi sifat fraktal yang dimilikinya, diharapkan mampu untuk memperbaiki kinerja dari perangkat lunak data mining yang telah dibuat dalam penelitian sebelumnya.

Dalam makalah ini satu studi perbandingan yang melibatkan kedua metode tersebut di atas dibahas. Perbandingan dilakukan baik terhadap kualitas pola-pola asosiasi yang dapat dibangkitkan oleh masing-masing metode maupun terhadap waktu komputasi yang diperlukan dalam proses ekstraksi pola-pola asosiasi yang diinginkan.

2. POLA ASOSIASI

Berdasarkan definisi dasar yang melatarbelakangi dibentuknya pola asosiasi dari suatu data transaksi *market basket*, pola asosiasi menggambarkan hubungan antar item data dimana jika suatu item dibeli dalam suatu transaksi, maka item yang lain juga dibeli. Untuk ini jika, X dan Y menyatakan dua kumpulan item, maka pola asosiasi dapat secara sederhana dirumuskan sebagai $X \rightarrow Y$, dimana kumpulan item X disebut sebagai *antecedent* (bagian yang mendahului) dan kumpulan item Y disebut sebagai *consequent* (bagian yang mengikuti) [9]. Sebagai satu contoh, jika dalam transaksi data suatu supermakert ditemukan bahwa seorang pelanggan yang membeli roti juga membeli mentega dan selai, maka dapat diperoleh satu kaidah asosiasi *roti* → *mentega, selai.*

Dalam penerapan metode graf asosiasi dan dimensi fraktal untuk pencarian pola-pola asosiasi, agar diperoleh pola-pola asosiasi yang berkualitas dan berguna, maka diperlukan sejumlah parameter untuk mengukur kualitas pola asosiasi yang diinginkan. Dalam penelitian yang dilakukan, terdapat empat parameter yang digunakan dalam melakukan proses pencarian pola-pola asosiasi, yaitu *minimum support*, *minimum* confidence, window *support*, dan *threshold fraktal*. Dua parameter yang pertama digunakan baik untuk metode graf asosiasi maupun metode dimensi fraktal, sedang dua parameter yang terakhir hanya digunakan untuk metode dimensi fraktal. Untuk sekumpulan item X sebagai *antecedent* serta sekumpulan item Y sebagai *consequent*, ketiga parameter pertama dapat didefinisikan seperi berikut.

Minimum support (*MinSup*) didefinisikan sebagai jumlah itemsets dari pola yang muncul dibagi dengan jumlah transaksi yang ada, yaitu:

$$MinSup = \frac{|X \cap Y|}{|D|} \qquad (1)$$

Minimum Confidence (*MinConf*) dari pola $X \rightarrow Y$ didefinisikan sebagai prosentase antara support dari transaksi yang di dalamnya mengandung X (dan juga mengandung Y)dan support dari X, yaitu

$$MinConf = \frac{Sup(X \cup Y)}{Sup(X)} \qquad (2)$$

Window Support (*WinSup*) pada dasarnya serupa dengan parameter minimum support, hanya saja parameter ini digunakan dalam kontek dimana basis data dibagi menjadi beberapa interval transaksi. Jadi, *WinSupport* dapat didefinisikan sebagai jumlah itemsets yang muncul dibagi dengan jumlah *interval* transaksi, yaitu

$$WinSup. = \frac{|X \cap Y|}{Interval} \qquad (3)$$

Threshold Fraktal (*Threshold*) didefinisikan sebagai nilai ambang batas fraktal (antara 0 dan 1) yang harus dipenuhi oleh basis data yang digunakan.

3. METODE GRAF ASOSIASI

Dalam pencarian pola-pola asosiasi yang didasarkan pada metode graf asosiasi, spesifikasi pola asosiasi dibedakan menjadi tiga tipe sesuai dengan struktur yang ditunjukkan dalam gambar 1. Spesifikasi tipe I meliputi semua spesifikasi yang tidak melibatkan simbol bintang dalam yang mengikuti bagian *antecedent* dan *consequent*. Spesikasi tipe II meliputi semua spesifikasi yang melibatkan tanda bintang pada bagian *antecedent* atau *consequent*. Sedang spesifikasi tipe III meliputi semua spesifikasi yang pada bagian *antecedent* maupun *consequent* hanya terdiri dari bintang saja.

Mining Association Rules
From <database>
With
Antecedent <items> [*]
Consequent <items> [*]
Support s%
Confidence c%

Gambar 1. Struktur Spesifikasi Pola Asosiasi

3.1 Algoritma untuk Memperoleh Pola Asosiasi

Dalam metode yang didasarkan pada graf asosiasi, algoritma untuk memperoleh pola asosiasi dibagi menjadi lima langkah utama, yaitu pembangkitan *bit vectors*, pembangkitan kumpulan item untuk spesifikasi tipe I, pembangkitan asosiasi graf, pembangkitan kumpulan item untuk spesifikasi tipe II dan III, dan pembangkitan pola-pola asosiasi.

Pembangkitan Bit Vectors

Merupakan langkah pertama yang harus dilakukan untuk memperolah sekumpulan item besar dimana pola-pola asosiasi yang potensial mungkin dapat diekstrak. Dalam langkah ini, basis data yang digunakan ditelusuri untuk mengidentifikasi item-item yang diinginkan sesuai dengan spesifikasi yang diberikan. Sebuah *bit vector* dengan panjang sama dengan jumlah transaksi yang ada dalam basis data dibuat untuk setiap item yang dinyatakan dalam spesifikasi. Bit ke-i dari *bit vector* diset dengan nilai "1" jika baris ke-i dari transaksi berisikan item yang dimaksud, dan diset dengan nilai "0" jika tidak.

Jika sebuah *bit vector* yang berkorespondensi dengan sebuah item x dinotasikan sebagai B_x, maka *support* untuk item x sama dengan jumlah nilai digit "1" dalam B_x. Dengan menggunakan definisi ini, *support* dari sekumpulan item $i_1, i_2, \ldots, i_k$ dapat diperoleh dengan melakukan operasi *inner product* ke semua *bit vector* yang bersesuaian, yaitu $B_1 \bullet B_2 \bullet \ldots\ldots \bullet B_k$.

Pembangkitan Kumpulan Item untuk Spesifikasi Tipe I

Kumpulan item besar untuk tipe spesifikasi I dapat secara sederhana diperoleh dengan melakukan *inner product* pada semua *bit vector* dari semua item yang dinyatakan dalam bagian *antecedent* dan *consequent* dari struktur spesifikasi yang ditentukan. Pola-pola asosiasi yang diekstrak dalam langkah ini adalah pola-pola asosiasi yang memenuhi batasan support minimum yang dinyatakan dalam spesifikasi.

Pembangkitan Graf Asosiasi

Langkah ini secara spesifik diperlukan sebelum proses pembangkitan item besar untuk spesifikasi tipe II dan III. Sebelum langkah ini digunakan, semua item dalam basis data diurut sedemikian rupa sehingga setiap item mempunyai nomor urutan yang unik.

Graf yang dibangkitkan berupa graf berarah. Simpul-simpul dari graf berkorespondensi dengan item-item yang harus dilibatkan sesuai dengan spesifikasi yang diinginkan. Sedang busur-busur dari graf menyatakan kumpulan item-item besar (selanjutnya akan disingkat sebagai *KIB*) yang terdiri dari 2 item (selanjutnya akan disebut sebagai *KIB-2*). Dalam kontek ini, KIB-2 berkorespondensi dengan sepasang item yang hasil operasi inner product dari kedua *bit vectors*-nya lebih besar atau sama dengan nilai batasan support minimum yang ditentukan dalam spesifikasi.

Dalam graf asosiasi, sebuah KIB-2 yang berkorespondensi dengan sepasang item i dan j dinyatakan sebagai sebuah busur berarah dari item i ke item j jika dan hanya jika nomor urutan posisi item i lebih kecil dari pada nomor urutan item j. Jika tidak, maka sebuah busur berarah dari item j ke item i harus dibuat. Gambar 2 berikut memperlihatkan sebuah graf asosiasi untuk sepesifikasi yang bagian *antecedent* dan *consequent*-nya berturut-turut diset dengan {C*} dan {E*} (dengan nilai support minimum diset sebesar 20%) dari sebuah contoh basis data sederhana yang ditunjukkan dalam tabel 1.

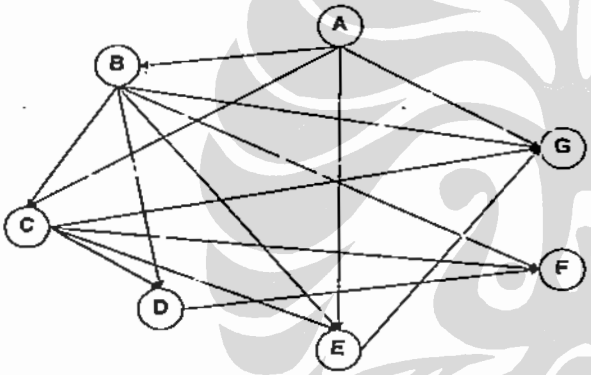
Pembangkitan KIB untuk Spesifikasi Tipe II

Untuk spesifikasi pola asosiasi tipe II, semua item yang ada dalam daftar item yang mengikuti bagian *antecedent* dan *consequrnt* harus dipertimbangkan dalam mencari semua KIB yang mungkin dapat diekstrak. Untuk ini, algoritma *Large Itemset*

Generation by Tree Expansion (*LGTE*) [8] dilibatkan dalam melakukan pencarian KIB.

Tabel 1. Contoh Transaksi Basis Data Sederhana

TID	Items-set
1	CEAGB
2	BDAECG
3	ABCEG
4	ECGA
5	GEACD
6	EGAH
7	AEG
8	AG
9	DFBC
10	BDFCH
11	CBF
12	BC
13	DFB
14	FB
15	CE



Gambar 2. Contoh Graf Asosiasi

Dengan menggunakan graf asosiasi yang telah diperoleh sebelumnya, LGTE akan membentuk sebuah *extended tree*. Dalam hal ini, LGTE akan berupaya untuk mengembangkan setiap simpul anak dari tree yang berisikan KIB-k ($k \geq 2$) guna memperoleh simpul-simpul anak lebih lanjut yang berisikan item k+1. Di bawah ini diberikan langkah-langkah utama dari algoritma LGTE dengan menggunakan graf asosiasi yang telah dihasilkan sebelumnya.

a) Dapatkan *FirstLargeItem* yang terdiri dari item-item yang disebutkan dalam bagian *antecedent* dan diurut secara menanjak berdasarkan urutan posisinya dalam basis data.
b) Periksa apakah terdapat satu lintasan yang menghubungkan item pertama ke item terakhir dalam *FirstLargeItem* dalam graf, dan juga periksa apakah nilai operasi *inner product* dari semua bit vector dari item-item yang bersesuaian memenuhi nilai support minimum. Jika tidak ditemukan lintasan yang dimaksud, hentikan proses (yang berarti bahwa *tree extension* tidak mungkin diperoleh). Jika tidak, buat tree dengan root berisikan semua item dalam *FirstLargeItem*, dan lanjutkan proses ke langkah (c).
c) Kembangkan *FirstLargeItem* untuk menghasilkan satu set item-item yang lebih besar (*NextLargeItems*), yaitu simpul-simpul anak kedua dari tree dengan cara berturut-turut memasukkan setiap item yang posisinya (i) lebih kecil dari item pertama dalam *FirstLargeItem*, (ii) berada di antara dua item yang ada dalam, dan (iii) lebih besar dari posisi item terakhir dalam *FirstLargeItem*.
d) Untuk setiap *NextLargeItem* yang ditemukan dalam langkah (c), lakukan proses *inner product* terhadap semua *bit vector* pada setiap item dalam *NextLargeItem* yang bersesuaian. Jika kardinalitas dari proses *inner product* memenuhi *minimum support* yang ditentukan, tambahkan *NextLargeItem* sebagai anak berikutnya dalam tree.
e) Ulangi langkah (c) hingga (d) untuk setiap *NextLargeItem* dengan menset *NextLargeItem* tersebut sebagai satu *FirstLargeItem* baru sehingga tidak ditemukan lagi simpul anak untuk tingkat berikutnya.

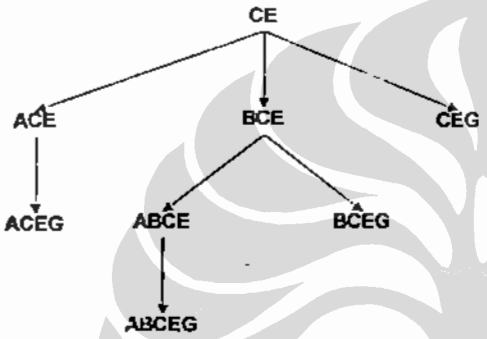
Jika algoritma di atas dikenakan terhadap graf asosiasi yang ditunjukkan dalam gambar 2, maka sebuah tree ekstensi seperti ditunjukkan dalam gambar 3 akan dihasilkan.

Pembangkitan KIB untuk Spesifikasi Tipe III

Untuk spesifikasi pengguna tipe III, oleh karena tidak ada item yang spesifik yang dinyatakan dalam parameter-parameter baik yang mengikuti bagian antecedent dan consequent, maka semua pola asosiasi yang memenuhi batasan *minimum support* dan *minimum confidence* harus dihasilkan. Untuk keperluan ini, algoritma *Large Itemset Generation by Direct Extension* (*LGDE*) yang merupakan ekstensi dari algoritma LGTE yang dijelaskan sebelumnya. Tetapi, sebagai pengganti pembatasan proses pembentukan pengembangan tree ke hanya satu set item yang dinyatakan dalam parameter <items> yang mengikuti bagian *antecedent* dan *consequent*, LGDE dapat menghasilkan tree ekstensi yang berlipat. Di

bawah ini dijelaskan rangkuman langkat-langkah utama yang dilibatkan dalam LGDE.

a) Set nilai awal *LargeItemsSet* = ∅
b) Untuk setiap KIB-2 yang ditemukan dalam langkah pembangkitan graf asosiasi, buat tree ekstensi dengan menggunakan item-item yang ada dalam KIB-2 yang sebelumnya sudah diurut secara menanjak berdasarkan nomor urutan item.
c) Untuk setiap item yang ada dalam KIB-2 yang diperoleh dalam langkah (b), dapatkan satu set item-item baru *NewLargeItemsSet* dengan menggunakan langkah (c) dan (d) dari algoritma LGTE yang dijelaskan sebelumnya. Tambahkan satu set item baru tersebut ke *LargeItemsSet* dengan melakukan operasi set union: *LargeItemsSet* = *LargeItemsSet* ∪ *NewLargeItemsSet*.



Gambar 3. Contoh sebuah Tree Ekstensi

Pembangkitan Pola Asosiasi

Pola-pola asosiasi yang diperoleh dari ketiga langkah yang dijelaskan sebelumnya hanya mempertimbangkan batasan *minimum support* yang harus dipenuhi. Pola-pola asosiasi ini harus diproses lebih lanjut guna menghasilkan pola-pola asosiasi akhir yang memenuhi batasan *minimum confidence* yang ditentukan oleh pengguna.

Seperti dijelaskan sebelumnya, perhitungan faktor *confidence* untuk sebuah pola sosiasi harus memperhatikan nilai support dari item-item yang mengikuti bagian *antecedent* dan *consequent* dari spesifikasi yang diberikan oleh pengguna. Sebagai contoh, jika notasi $x \rightarrow y$ menyatakan satu pola asosiasi dengan x dan y masing-masing merepresentasikan satu set item yang mengikuti bagian *antecedent* dan *consequent*, maka faktor *confidence* dari pola asosiasi tersebut dapat dinayatakan sebagai pembagian dari support untuk x dan y dibagi dengan nilai suport dari x saja.

Untuk spesifikasi pengguna tipe I, nilai faktor *confidence* dapat diperoleh secara langsung seeperti dicontohkan sebelumnya. Untuk spesifikasi tipe II, faktor *confidence* dari setiap pola asosiasi harus dihitung dan dibandingkan dengan faktor *confidence* yang diberikan dalam spesifikasi. Sebagai contoh, jika KIB berupa {A,C,E} ditemukan dalam suatu spesifikasi pengguna tipe II, yang secara simbolik dapat dituliskan sebagai C* → E*, maka faktor-faktor *confidence* dari pola asosiasi AC→ E and C → AE harus juga dievaluasi. Hanya pola-pola sosiasi yang memnuhi nilai minimum faktor confidence yang dipilih sebagai hasil. Cara yang sama dapat digunakan untuk spesifikasi yang berisikan kebutuhan pengguna tipe III. Untuk ini, perhitungan nilai faktor *confidence* harus dilakukan pada setiap kombinasi pola asosiasi yang mungkin dapat dihasilkan dari satu set KIB yang dihasilkan.

4. DIMENSI FRAKTAL

Fraktal merupakan sebuah bentuk kasar dari obyek geometri yang dapat terbagi menjadi bagian-bagian yang lebih kecil namun memiliki sifat kemiripan diri (*self similarity*) terhadap obyek asalnya dengan penskalaan yang bebas [1,2]. Sebagai suatu ilustrasi, sebuah pohon dapat memiliki sejumlah cabang yang besar, dimana setiap cabang ini dapat mempunyai beberapa cabang yang lebih kecil (ranting). Setiap ranting, pada gilirannya, dapat memiliki sejumlah ranting yang lebih kecil lagi, demikian seterusnya sehingga semua obyek cabang pohon tersebut mempunyai bentuk yang kelihatannya sama dengan ukuran yang berbeda.

Obyek fraktal mempunyai dua karakteristik dasar, yaitu tingkat kedetilan yang tidak terbatas pada setiap titiknya dan sifat kemiripan diri antara bagian suatu obyek dan bentuk obyek tersebut. Sifat kemiripan diri dari sebuah obyek dapat dilihat dari berbagai bentuk, tergantung pada representasi obyek itu sendiri. Sebuah obyek fraktal dapat dispesifikasikan dengan operasi berulang untuk menghasilkan bagian-bagian yang lebih rinci dari sebuah obyek yang disajikan.

Setiap bagian dari obyek fraktal merupakan penskalaan dari obyek sebelumnya. Dengan demikian, dari sebuah obyek awal, dapat dibangun sub-bagian dari obyek dengan penskalaan sebesar s yang lebih kecil dari pada obyek asal. Selama proses iterasi dijalankan, faktor skala s yang sama atau berbeda dapat digunakan untuk menghasilkan obyek-obyek fraktal yang berbeda.

Pengertian *"dimensi"* dalam dimensi fraktal menyatakan orde atau derajad dari geometri. Secara tradisional, sebuah bentuk geometri dapat mempunyai dimensi 0 (titik), dimensi 1 (garis), dimensi 2 (bidang), atau dimensi 3 (ruang). Secara

teoritis, dimensi-dimensi tersebut dapat diperluas hingga meliputi dimensi yang lebih tinggi [1,2].

Banyaknya variasi dari sebuah obyek fraktal dapat dijelaskan dengan menggunakan sebuah bilangan dimensi fraktal (bilangan D), yang menyatakan ukuran dari obyek tersebut. Untuk ini, semakin sebuah obyek terlihat bergerigi, maka obyek tersebut akan mempunyai nilai dimensi fraktal yang lebih besar. Secara umum, rumusan dari dimensi fraktal dapat dituliskan seperti persamaan berikut [1]:

$$D_q = \frac{1}{q-1}\frac{\partial \log \sum_i p_i^q}{\partial \log r} = Constant;\ q \neq 1 \text{ dan } r_1 < r < r_2 \quad (4)$$

$$D_0 = \frac{-\partial \log(N(r))}{\partial \log r} = Constant; r_1 < r < r_2 \quad (5)$$

$$D_1 = \frac{\partial \log \sum_i p_i \log p_i}{\partial \log r} = Constant;\ r_1 < r < r_2 \quad (6)$$

$$D_2 = \frac{\partial \log \sum_i p_i \log p_i^2}{\partial \log r} = Constant;\ r \in (r_1, r_2) \quad (7)$$

Untuk semua persamaan di atas, q menyatakan variabel dimensi fraktal, dimana untuk $q = 0$, 1 dan 2 berturut-turut disebut sebagai *hausdorf fractal dimension*, *information fractal dimension*, dan *correlation fractal dimension* [5]. Notasi p dan r, masing-masing menyatakan nilai total sel dari sebuah ruang dan jumlah titik dari setiap window yang digunakan. Notasi $N(r)$ menyatakan jumlah itemsets yang memenuhi *WinSup*, *MinSup*, dan *MinConf*. Sedang r_1 dan r_2, berturut-turut menyatakan batas atas dan batas bawah dari sebuah interval suatu window.

5. PENGGUNAAN DIMENSI FRAKTAL DALAM PENCARIAN POLA ASOSIASI

Seperti dijelaskan sebelumnya bahwa tujuan utama dari perangkat lunak data mining yang dibuat adalah berupaya untuk menemukan pola-pola asosiasi yang berkualitas dan bermanfaat. Penggunaan metode dimensi fraktal dalam data mining diharapkan dapat memberikan satu alternatif solusi yang cepat dalam menemukan pola-pola asosiasi dari suatu basis data yang besar. Ide dasar dari metode dimensi fraktal ini adalah memodifikasi algoritma apriori yang telah banyak digunakan dalam data mining. Dalam penerapannya, metode dimensi fraktal secara iteratif akan membagi basis data menjadi bagian-bagian yang lebih kecil sehingga dapat ditemukan pola asosiasi dalam jumlah yang lebih besar dan lebih bermanfaat bagi penggunaya.

5.1 Algoritma Apriori

Algoritma ini merupakan bagian yang terpenting dari keseluruhan algoritma fraktal. Pada saat ini telah banyak dikembangkan modifikasi algoritma apriori untuk menangani proses pencarian dari sekumpulan data dalam jumlah yang besar. Algoritma apriori merupakan salah satu algoritma yang paling populer dalam keseluruhan proses pencarian pola asosiasi yang dilibatkan dalam data mining [3,4]. Gambar 4 memperlihatkan *pseudo-code* dari "algoritma apriori". Sedang *psedudo-code* fungsi "apriori-gen" yang dipanggil dalam algoritma apriori ditunjukkan dalam gambar 5. Dalam kedua gambar ini, notasi-notasi k-itemset, L_k dan C_k berturut-turut menyatakan itemset yang memiliki k-items, kumpulan dari k-itemset yang besar (large itemsets) dan kumpulan dari kandidat k-itemset.

```
1)  L1 = {Large-1 itemsets}
2)  FOR (k=2; Lk-1 ≠ 0; k++) DO
3)  BEGIN
4)    Ck= apriori-gen(Lk-1); //new candidates
5)    FORALL transactions t ∈ D DO
6)    BEGIN
7)      Ct= subset(Ck,t); //candidates contained
      in t
8)      FORALL candidates c ∈ Ct DO
9)        c.count++;
10)   END
11)   Lk= {c ∈ Ck | c.count ≥ MinSup}
12) END
13) Aswer = ∪k Lk
```

Gambar 4. *Pseudo-code* Algoritma Apriori

```
1) INSERT INTO Ck
2)   SELECT P.item1,...., P.itemk-1, Q.itemk-1
3)   FROM Lk-1 P, Lk-1 Q
4)   WHERE P.item1=Q.item1,....,P.itemk-
     2=Q.itemk-2,
          P.itemk-1<Q.itemk-1
   Selanjutnya dilakukan proses pemotongan
   dengan menghapus semua itemsets c ∈ Ck,
   sehingga beberapa subset (k-1) tidak
   menjadi anggota Lk-1
5) FORALL itemsets c ∈ Ck DO
6)   FORALL (k-1) subsets s of c DO
7)     IF s ∉ Lk-1 THEN
8)       DELETE c FROM Ck
```

Gambar 5. *Pseudo-code* Fungsi Apriori-Gen

Seperti terlihat dalam gambar 4, proses pencarian *large itemset* (L_k) dilakukan secara iteratif. Akses basis data hanya dilakukan satu kali, dan semua *large itemset* yang dihitung harus diurutkan secara menanjak. Pada iterasi pertama, *large itemsets* diperoleh dengan melakukan *"scanning"* basis data sebanyak satu kali saja. Selanjutnya, untuk iterasi ke k ($k > 1$), kumpulan dari kandidat C_k dibentuk dengan menggunakan fungsi pencarian kandidat **"apriori-gen"** seperti ditunjukkan dalam gambar 4. Untuk ini L_{k-1} menyatakan kumpulan itemsets *k-1* yang ditemukan pada iterasi ke k-1.

5.2 Algoritma Fraktal

Algoritma ini merupakan modifikasi terhadap algoritma apriori yang dijelaskan dalam bagian sebelumnya. Modifikasi dilakukan mulai baris ke-5. Secara keseluruhan, rangkuman dari algoritma fraktal untuk pencarian pola-pola asosiasi yang telah berhasil diimplementasikan dapat dilihat dalam gambar 6.

```
1)   L1 = {Large 1-itemsets of categorical values}
2)   FOR (k=2; Lk-1!=0; k++) DO
3)   BEGIN
4)     Ck = generate(Lk-1);
5)     FOR every i in Ck DO
6)     BEGIN
7)       Make Mi = 0;
8)       Count = 0;
9)       Make lowi = interval(t0);
10)      twindowi = 0;
11)      WHILE there are transactions in D DO
12)      BEGIN
13)        Consider next transactions t in D;
14)        twindowi++;
15)        FOR every candidate j in Ck DO
16)        BEGIN
17)         Make upj = interval(t);
18)         IF j is in t THEN
19)         BEGIN
20)           j.count ++; j.Mcount++;
21)           Add row with interval(t),in(t,j) to Mj
22)         END;
23)         Compute F(Mj) = fd(Mj);
24)         IF (a change in F(Mj)) ≥ τ THEN
25)         BEGIN
26)           IF j.Mcount/D >= MinSup OR
                 j.Mcount/twindow ≥ MinSup
THEN
27)             Output itemset [lowj,upj],j;
28)           Count=0;
29)           j.Mcount = 0;
30)           lowj=interval(t);
31)           twindowj=0;
32)         END;
33)       END;
34)     END;
35)     Lk = {i in Ck | i.count ≥ MinSup};
36)  END;
37)  Output items in every Lk;
```

Gambar 6. *Pseudo-code* Algoritma Fraktal

Seperti diperlihatkan dalam gambar 6, baris ke-1 hingga ke-4 sama persis dengan algoritma apriori. Fungsi "generate" dalam algoritma fraktal pada dasarnya serupa dengan fungsi "apriori-gen" seperti ditunjukkan dalam gambar 5. Fungsi ini ditujukan untuk memperoleh kandidat awal dari pola-pola asosiasi pada setiap iterasi.

Tahap inisialisasi dilakukan pada baris 7 sampai dengan 9. Baris ke-7 merupakan proses iterasi untuk semua item pada kandidat yang diperoleh sebelumnya. Proses iterasi ini dilakukan jika suatu item memiliki count ≥ MinSup. Baris ke-7 dan ke-8 variabel M_i dan Count, masing-masing diset dengan nilai 0 untuk semua item. Variabel M_I ini digunakan untuk menampung banyaknya transaksi yang sesuai dalam suatu interval, sedangkan Count digunakan untuk menampung banyaknya transaksi yang sesuai dari seluruh basis data yang digunakan. Selanjutnya variabel diset dengan low = interval(t_0), yang berfungsi untuk menampung posisi interval yang paling rendah. Selain itu, juga dilakukan pengesetan twindow = 0, yang berfungsi untuk menampung banyaknya transaksi yang sesuai maupun tidak.

Inti dari algoritma fraktal ditunjukkan dalam baris 11 sampai dengan 34. Dalam inti algoritma ini dilakukan peremajaan terhadap nilai-nilai parameter yang berhubungan dengan data transaksi. Untuk setiap iterasi dilakukan penambahan nilai twindow (baris 14) dan penentuan nilai variabel up = posisi transaksi pada saat itu. Variabel up merupakan parameter yang berfungsi untuk menampung nilai posisi transaksi bagian atas dari suatu interval window tertentu. Jika kemudian ditemukan transaksi yang mengandung item yang dimaksud, maka dilakukan penambahan nilai count dan Mcount (baris 20). Kedua parameter count dan Mcount digunakan untuk menghitung jumlah transaksi yang sesuai dengan itemsets yang dimaksud.

Seperti dijelaskan sebelumnya bahwa, perhitungan dimensi fraktal yang digunakan dalam algoritma fraktal untuk pencarian pola-pola asosiasi didasarkan pada mekanisme *Hausdorf fractal dimension* (persamaan 5). Penurunan (differensial) dari persamaan 5 dapat dinyatakan dalam bentuk limit seperti berikut:

$$D_0 = -\lim_{r \to 0} \frac{\log(N(r))}{\log r} \qquad (8)$$

Dari persamaan di atas terlihat bahwa nilai fraktal untuk masing-masing transaksi dapat diperoleh dari

perbandingan logaritma $N(r)$ dengan r, dimana dalam algoritma fraktal r direpresentasikan dengan `Mcount`, sedang $N(r)$ direpresentasikan dengan `twindow`. Dalam perhitungan pencarian nilai fraktal, D_0 diperoleh dengan menghitung nilai absolutnya.

Setelah diperoleh fraktal yang diinginkan, maka fraktal item tersebut dibandingkan dengan nilai ambang batas (*threshold*) dari fraktal (τ) yang bernilai antara 0 dan 1. Jika fraktal yang diperoleh pada saat transaksi berlangsung lebih kecil dibandingkan dengan nilai ambang batas fraktal, maka dapat disimpulkan bahwa tidak ada kemiripan (*unsimilarity*). Sebaliknya jika didapatkan fraktal yang lebih besar atau sama dengan nilai ambang batas fraktal, maka dapat disimpulkan terdapat kemiripan (*similarity*). Jika terdapat kemiripan maka dilakukan pengecekan nilai `count` dan `Mcount` terhadap `MinSup` dan `WinSup` (baris 26). `MinSup` dan `WinSup` adalah parameter yang nilainya ditentukan oleh pengguna. Jika `Mcount` ≥ `MinSup` atau `count` ≥ `WinSup` maka item yang diperoleh akan disimpan dalam array (baris 27), dan kemudian dilakukan pengesetan nilai `Mcount=0`, dan `low=interval(t)` dan `twindow=0` (baris 28-31).

6. HASIL UJI COBA

Kedua perangkat lunak yang didesain dan diimplementasikan untuk dijalankan dalam lingkungan sistem operasi Windows telah menunjukkan kapabilitasnya dalam memperoleh kaidah asosiasi yang berkualitas sesuai dengan batasan-batasan yang telah ditetapkan. Untuk membandingkan kinerja dari kedua perangkat lunak, digunakan satu set data uji coba yang terdiri tiga jenis data sintesis (tabel 2) yang dihasilkan oleh perangkat lunak pembangkit data berbasis skenario [3] yang dibuat khusus untuk kebutuhan aplikasi data mining. Semua uji coba dilakukan dalam lingkungan Windows NT yang dijalankan di atas komputer PC Pentium II (HP-Vectra) 450 MHz dan memory sebesar 128 MB.

Table 2. Spesifikasi Data Uji Coba

	Data-1	Data-2	Data-3
Jumlah Item	25	25	50
Jumlah Transaksi	1000	5000	10064
Jumlah Record	13758	41131	81922

Untuk mendapatkan hasil perbandingan yang konsisten dari kedua perangkat lunak, nilai parameter untuk *minimum support* dan *minimum confidence* dibuat sama, karena kedua perangkat lunak mempunyai cara pengukuran yang sama untuk kedua parameter tersebut. Namun demikian, untuk perangkat lunak yang didasarkan pada penerapan dimensi fraktal, nilai dari dua parameter tambahan, yaitu *window support* dan *ambang batas fraktal* harus juga diberikan.

Perbandingan kinerja dari kedua perangkat lunak diukur berdasarkan jumlah pola asosiasi dan waktu komputasi yang diperlukan untuk spesifikasi yang sama. Tabel 3 sampai dengan tabel 5 memperlihatkan hasil perbandingan dari kedua perangkat lunak, berturut-turut untuk data uji coba Data-1, Data-2 dan Data-3. Untuk setiap tabel hasil uji coba, nilai-nilai yang ditempatkan dalam kolom dengan nama "GA" dan "DF", berturut-turut merupakan hasil yang diperoleh untuk perangkat lunak yang didasarkan pada algoritma "Graf Asosiasi" dan algoritma "Dimensi Fraktal".

Dari hasil uji coba untuk semua jenis data yang digunakan terlihat bahwa kinerja perangkat lunak yang didasarkan pada algoritma dimensi fraktal secara signifikan lebih baik dibandingkan dengan kinerja perangkat lunak yang didasarkan pada algoritma graf asosiasi. Dengan menggunakan nilai minimum support dan minimum *confidence* yang sama untuk kedua perangkat lunak, jumlah asosiasi yang diperoleh pada perangkat lunak yang didasarkan pada algoritma dimensi fraktal jauh lebih banyak dibandingkan dengan perangkat lunak yang didasarkan pada algoritma graf asosiasi. Keadaan yang sama terjadi pada waktu komputasi yang diperlukan oleh kedua algoritma, dimana waktu yang diperlukan oleh perangkat lunak yang didasarkan pada algoritma dimensi fraktal 3 hingga 10 kali lebih cepat dibandingkan dengan perangkat lunak yang didasarkan pada algoritma graf asosiasi.

Kelebihan kinerja perangkat lunak yang didasarkan pada algoritma dimensi fraktal terutama mungkin disebabkan oleh adanya keunggulan dalam melakukan partisi dari seluruh transaksi berdasarkan nilai kemiripan dimensi fraktal yang dimilikinya. Dengan cara ini memungkinkan algoritma dimensi fraktal untuk melakukan daerah pelacakan yang lebih baik dibandingkan dengan algoritma yang didasarkan pada graf asosiasi.

Tabel 3. Perbandingan Hasil Uji Coba dengan menggunakan Data-1

Parameter	Hasil Uji Coba	
Antecedent	item1	
Cosequent	item2	
	GA	DF
Minimum Support	30%	30%
Minimum Confidence	50%	50%
Window Support	-	30%
Ambang Batas Fraktal	-	0,5
Jumlah Pola	1	47
Waktu (detik)	6,67	2,25

Tabel 4. Perbandingan Hasil Uji Coba dengan menggunakan Data-2

Parameter	Hasil Uji Coba			
Antecedent	item13		item13, item14	
Cosequent	item15		item15, item20	
	GA	DF	GA	DF
Minimum Support	20%	20%	20%	20%
Minimum Confidence	50%	50%	50%	50%
Window Support	-	20%	-	20%
Ambang Batas Fraktal	-	0,5	-	0,5
Jumlah Pola	1	135	1	106
Waktu (detik)	30,263	9,643	35,88	10,164

Tabel 5. Perbandingan Hasil Uji Coba dengan menggunakan Data-3

Parameter	Hasil Uji Coba			
Antecedent	Item26		Item23, item26	
Cosequent	Item28		Item28, item29	
	GA	DF	GA	DF
Minimum Support	10%	10%	2%	2%
Minimum Confidence	30%	30%	15%	15%
Window Support	-	10%	-	2%
Ambang Batas Fraktal	-	0,5	-	0,5
Jumlah Pola	1	13	1	5
Waktu (detik)	68,92	6,14	98,55	9,89

7. KESIMPULAN

Makalah ini telah membahas perbandingan algoritma dari kedua perangkat lunak untuk mengekstraksi pola-pola asosiasi yang mungkin dapat ditemukan dalam suatu basis data. Kedua perangkat lunak yang dibandingkan masing-masing didasarkan pada algoritma graf asosiasi dan algoritma dimensi fraktal.

Hasil uji coba perbandingan menunjukkan bahwa kinerja dari perangkat lunak data mining dengan menggunakan metode dimensi fraktal secara signifikan lebih baik dibandingkan dengan kinerja

perangkat lunak yang didasarkan pada algoritma graf asosiasi. Keunggulan tersebut terjadi baik pada jumlah asosiasi yang dapat diekstrak dari basis data maupun waktu komputasi yang diperlukan dalam eksekusinya.

Selain itu, khusus untuk perangkat lunak yang didasarkan pada algoritma dimensi fraktal dapat disimpulkan bahwa penentuan nilai parameter ambang batas fraktal dapat memberikan pengaruh yang cukup signifikan terhadap jumlah pola asosiasi yang dapat diekstrak dari basis data. Untuk ini kecenderungan yang terjadi adalah semakin kecil nilai ambang batas yang ditentukan semakin besar jumlah pola asosiasi yang dapat diperoleh.

REFERENSI

[1] Alberto Belussi and Christolus Faloutsos, "Estimating the Selectivity of Spatial Queries using the Correlation Fraktal Dimension", *Proceeedings of 21st Int'l Conference on Very Large Databases*, September 1995, pp. 299-310.

[2] Arif Djunaidy, R. Soelaiman and P. Ananto, "Development of Data Mining Application Software for Exploiting User-Defined Association Rules", *Proceedings of the Int'l Conference on Electrical, Electronics, Communication, and Information (CECI'2001)*, Jakarta, March 2001.

[3] Arif Djunaidy, R. Soelaiman and D.H. Prasetyo, "Development of a Scenario-Based Data Generator for Data Warehousing and Data Mining Applications", *Proceedings of the Int'l Conference on Eletrical, Electronics, Communication, and Information (CECI'2001)*, Jakarta, March 2001.

[4] Arif Djunaidy, R. Soelaiman dan Eko B. Setiawan, "Penerapan Metode Dimensi Fraktal pada Data Mining untuk Penentuan Kaidah Asosiasi", *Proceedings Seminar Nasional Kecerdasan Komputasional II (SNKK II)*, UI Jakarta, Oktober 2001.

[5] Berry M.J. and G. Linoff, *Data Mining Techniques for Marketing, Sales, and Customer Support*, John Wiley and Sons, 1997.

[6] Dick Oliver, *Fraktal Vision: Put Fraktals to Work for You*, Sams, New York, 1992.

[7] K. Falconer, *Fraktal Geometry Mathematical Foundations and Applications*, John Wiley & Sons, New York, 1990.

[8] Meo R., G. Psaila and S. Ceri, "A New SQL-Like Operator for Mining Associations Rules", *Proceedings of the International Conference on Very Large Databases*, 1996.

[9] R. Agrawal et. al., "Mining Association Rules between Sets of Items in Large Databases", *Proceedings of the ACM SIGMOID*, Washington, May 1993.

[10] R. Agrawal R. and R. Srikant, "Fast Algorithm for Mining Association Rules", *Proceeedings of 20th Int'l Conference on Very Large Databases*, Santiago, Chile, 1995.

[11] Yen S-J and Arbee L.P. Chen, "An Efficient Approach for Discovering Knowledge from Large Databases", *Proceedings of the 8th International Conference and Workshop on Database and Expert Systems Applications*, 1997.