Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

**Vol. 5, No. 3, Januari 2010** **ISSN 0216 - 0544**

# DETEKSI *OUTLIER* BERBASIS KLASTER PADA SET DATA DENGAN ATRIBUT CAMPURAN NUMERIK DAN KATEGORIKAL

**[*]Dwi Maryono, [**]Arif Djunaidy**
Program Magister Teknik Informatika, Fakultas Teknologi Informasi, ITS
Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111
E-Mail: [*]wimar@cs.its.ac.id, [**]adjunaidy@its.ac.id

**Abstrak**

Deteksi *outlier* merupakan salah satu bidang penelitian yang penting dalam topik data *mining*. Penelitian ini bermanfaat untuk mendeteksi perilaku yang tidak normal seperti deteksi intrusi jaringan, diagnosa medis, dan lain-lain. Banyak metode telah dikembangkan untuk menyelesaikan masalah ini, namun kebanyakan hanya fokus pada data dengan atribut yang seragam, yaitu data numerik atau data kategorikal saja. Kenyataan di lapangan, set data seringkali merupakan gabungan dari dua nilai atribut seperti ini. Dalam penelitian ini diajukan sebuah metode untuk mendeteksi *outlier* pada set data campuran yaitu *MixCBLOF*. Algoritma ini merupakan gabungan dari beberapa teknik, seperti klasterisasi subset data, deteksi *outlier* berbasis klaster, dan penggunaan *Multi-Atribute Decision Making* (*MADM*). Uji coba dilakukan pada beberapa set data dari *UCI Machine Learning Repository*. Evaluasi dilakukan dengan membandingkan rata-rata pencapaian *coverage* untuk *top ratio* antara jumlah *outlier* eksak dengan jumlah data. Dari uji coba yang dilakukan, diperoleh hasil bahwa *MixCBLOF* cukup efektif untuk mendeteksi *outlier* pada set data campuran dengan rata-rata pencapaian *coverage* 73,54%. Hasil ini lebih baik dibandingkan dengan algoritma *CBLOF* yang diterapkan pada set data yang telah didiskritisasi dengan rata-rata pencapaian *coverage* 67,98%, untuk diskritisasi dengan *K-Means*, dan 59,48% untuk diskritisasi dengan *equal width*.

Kata kunci: data campuran, deteksi *outlier*, *Outlier* berbasis klaster, *CBLOF*, *MixCBLOF*.

***Abstract***

*Outlier detection is one of most the important research on mining data. This data is useful to detect abnormal behaviour such as networking detection, medical diagnosis and the others. Such methods have been developed to solve these problems, yet mostly focus on the data in similar attribute like numerical and categorical. Set data, in fact, is combination of the two attributes. This research purposes a method to detect the outlier at mix data set, like Mix CBLOF. Furthermore, algorithm is combination of several techniques such as subset cluster, outlier detection cluster based, and Multi-attribute Decision Making (MADM). A test was done of a set of data from UCI Machine Learning Repository. The Evaluation is conducted to compare the means of coverage achieiving for top ratio between the amount of exact outlier and the amount of data. From the test, it can be concluded that MixCBLOF is effective to detect outlier at set of mix data of means of coverage achieiving 73.54%. This result is better with CBLOF algorithm which is applied at the data set discridit with coverage achieiving 67.98% for discreet with K-Means, and 59.48% for equal width discreet.*

*Key words: mix data, outlier detection, outlier cluster based, CBLOF, MixCBLOF*

.

## PENDAHULUAN

Deteksi *outlier* pada sekumpulan data adalah salah satu bidang penelitian yang terus berkembang dalam topik data *mining*. Penelitian ini sangat bermanfaat untuk mendeteksi adanya perilaku atau kejadian yang tidak normal seperti deteksi penipuan penggunaan kartu kredit, deteksi intrusi jaringan, penggelapan asuransi, diagnosa medis, segmentasi pelanggan, dan sebagainya.

Bermacam-macam metode telah dikembangkan baik berdasarkan teknik ataupun jenis data yang dijadikan obyek. Untuk set data numerik, ada banyak teknik yang telah dikembangkan seperti *statistic-based*, *distance-based*, *density-based*, *clustering-based*, *subspace-based*, dan lain-lain. Sedangkan untuk set data kategorikal teknik yang dapat digunakan di antaranya adalah *CBLOF*, *FPOF* dan *LSA*. Namun demikian kebanyakan metode tersebut hanya fokus pada set data yang seragam, yaitu hanya terdiri dari salah satu tipe atribut saja. Adanya tipe atribut yang berbeda biasanya diatasi dengan melakukan transformasi dari salah satu tipe data menjadi tipe data yang lain, seperti diskritisasi atribut numerik. Namun demikian metode diskritisasi atribut numerik ini terdapat kekurangan seperti yang disebutkan Tan dkk [1]. Kekurangannya, antara lain adalah sulitnya menetapkan jumlah interval yang tepat sehingga dapat menyebabkan banyak pola yang *redundant* atau sebaliknya banyak pola yang hilang. Ini akan sangat berpengaruh jika atribut numerik cukup banyak dalam set data.

Sejauh ini tidak banyak penelitian yang bekerja pada data campuran seperti ini. He dkk [2] telah melakukan klasterisasi pada data campuran dengan pendekatan *divide and conquer*. Ia membagi set data menjadi dua *subset* data, yaitu numerik dan kategorikal. Masing-masing subset data diklasterisasi, kemudian hasilnya digabungkan. Data hasil penggabungan keduanya kemudian diklaster lagi untuk mendapatkan hasil akhir. Hasil eksperimen menunjukkan bahwa metode ini cukup efektif untuk melakukan klasterisasi.

Jika klasterisasi dapat dilakukan dengan partisi data numerik dan kategorikal [2], maka tentunya cara ini juga memungkinkan untuk deteksi *outlier*.

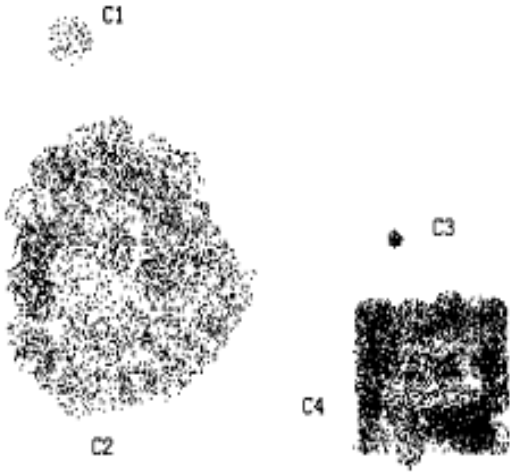
Mengingat penelitian lain juga menunjukkan bahwa deteksi *outlier* pada *subset* data tertentu dapat digunakan untuk mendeteksi *outlier* dari keseluruhan set data [3,4] Selain itu, penggabungan klasterisasi *subset* data juga digunakan untuk menemukan *outlier* pada data numerik dengan konsep *cluster uncertainty* [5].

Dari beberapa penelitian yang disebutkan di atas, dimungkinkan untuk melakukan beberapa pendekatan yang dapat diusulkan dalam penelitian ini. Di antaranya adalah pembagian set data menjadi numerik dan kategorikal, deteksi *outlier* pada *subset* data, dan pemanfaatan klasterisasi untuk untuk deteksi *outlier*. Untuk dapat menerapkan ide tersebut digunakan definisi *outlier* yang paling tepat. *Outlier* didefinisikan berbasis klaster, dimana sebuah *outlier* didefinisikan sebagai sembarang obyek yang tidak berada pada klaster yang ”cukup besar” [6]. Meskipun konsep ini diusulkan untuk data kategorikal, tapi sangat memungkinkan untuk diterapkan dengan data numerik dengan menggunakan konsep jarak.

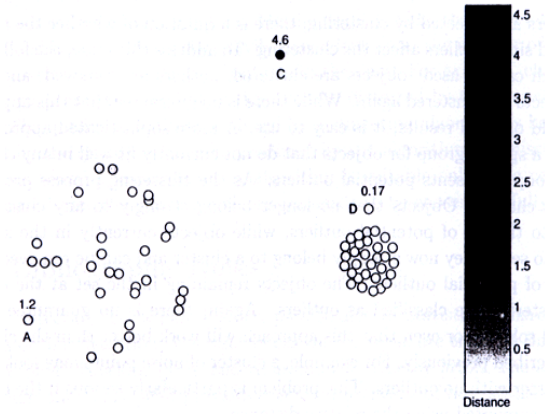
Penelitian ini dilakukan untuk menggabungkan beberapa pendekatan di atas dengan langkah-langkah sebagai berikut. Pertama, bagi set data menjadi dua bagian, yaitu *subset* data numerik dan kategorikal [2]. Selanjutnya dilakukan teknik klasterisasi dan deteksi *outlier* pada masing-masing partisi secara terpisah. Untuk meningkatkan hasil deteksi *outlier* pada keseluruhan data, dilakukan teknik persilangan. Hasil klasterisasi sub data numerik digunakan untuk menentukan derajat *outlier* berbasis klaster dengan atribut sub data kategorikal. Dan sebaliknya hasil klasterisasi sub data kategorikal digunakan untuk menentukan derajat *outlier* dengan menggunakan atribut numerik. Selanjutnya, untuk menggabungkan hasil langkah-langkah ini dapat digunakan *multi-atribut decision making* (*MADM*) yaitu dengan menggunakan fungsi atau operator *agregat* tertentu [7].

## DETEKSI *OUTLIER* BERBASIS KLASTER

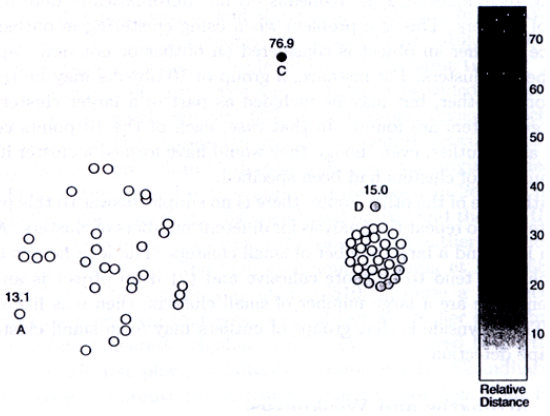
Metode yang diajukan dalam penelitian ini adalah pengembangan dari konsep *outlier* berbasis klaster yaitu dengan mendefinisikan konsep baru mengenai deteksi *outlier* berbasis klaster [6] (Gambar 1).




Gambar 1. Set Data DS1 [8].




Gambar 2. Jarak Obyek dari *Centroid* Terdekat [10].




Gambar 3. Jarak Relatif Obyek dari *Centroid* Terdekat [10].

Gambar 1 memperlihatkan data dua dimensi yang terdiri dari empat klaster $C_1$, $C_2$, $C_3$, dan $C_4$. Dari sudut pandang klaster, obyek-obyek data pada $C_1$ dan $C_3$ dapat dianggap sebagai *outlier* karena tidak terdapat pada klaster yang besar yaitu $C_2$ dan $C_4$. $C_2$ dan $C_4$ disebut klaster besar karena $C_2$ dan $C_4$ merupakan klaster yang dominan pada set data, yaitu memuat sebagian besar obyek pada set data [9].

Konsep *CBLOF* digunakan untuk menyelesaikan masalah deteksi *outlier* pada data kategorikal [6]. Namun, dalam penelitian ini dapat ditunjukkan bahwa konsep ini juga dapat dikembangkan untuk data numerik juga.

### *CBLOF* (*Cluster-Based Outlier Factor*)

Untuk mengidentifikasi signifikansi fisik dari definisi *outlier*, setiap obyek didefinisikan dengan sebuah derajat yang disebut dengan *CBLOF* (*Cluster Based Local Outlier Factor*) yang diukur dengan ukuran klaster di mana ia berada dan jaraknya terhadap klaster terdekat (jika ia terdapat dalam obyek kecil) [6].

Definisi 1**:** Misalkan $A_1$, $A_2$, ..., $A_m$ adalah himpunan atribut dengan domain $D_1$, $D_2$, ..., $D_m$. Set data $D$ terdiri dari *record* atau transaksi $t$: $t \in D_1 \times D_2 \times \ldots \times D_m$. Hasil klasterisasi pada $D$ dinotasikan sebagai $C = \{C_1, C_2, \ldots, C_k\}$ dimana $C_i \cap C_j = \varnothing$ dan $C_1 \cup C_2 \cup \ldots \cup C_k = D$, dengan $k$ adalah jumlah klaster.

Masalah yang penting pada tahap selanjutnya adalah pendefinisian klaster besar (*large cluster*) dan klaster kecil (*small cluster*).

Definisi 2**:** Misalkan $C = \{C_1, C_2, \ldots, C_k\}$ adalah himpunan klaster pada set data dengan urutan ukuran klaster adalah $|C1| \geq |C2| \geq \ldots \geq |Ck|$. Diberikan dua parameter numerik $\alpha$ dan $\beta$. Didefinisikan $b$ sebagai batas antara klaster besar dan kecil jika memenuhi formula pada Persamaan (1) dan (9).

$$(|C_1|+|C_2|+\ldots+|C_b|) \geq |D| * \alpha \quad (1)$$

$$|C_b|/|C_b+1| \geq \beta \quad (2)$$

Didefinisikan himpunan klaster besar (*large cluster*) sebagai $LC = \{C_i, / \ i \leq b\}$ dan klaster kecil (*small cluster*) didefinisikan dengan $SC = \{C_i, / \ i > b\}$.

Definisi 2 memberikan ukuran kuantitatif untuk membedakan klaster besar dan kecil. Persamaan (1) menunjukkan bahwa sebagian besar data bukan *outlier*. Oleh karena itu klaster besar mempunyai porsi yang jauh besar.

Sebagai contoh jika α diberikan 90% maka artinya lebih klaster besar memuat lebih dari 90% dari total obyek data pada set data. Persamaan (2) menunjukkan fakta bahwa klaster besar dan kecil harus memiliki perbedaan yang signifikan. Jika diberikan $\beta$ = 5, artinya setiap klaster besar minimal 5 kali lebih besar dari klaster kecil.

Definisi 3: Misalkan $C = \{C_1, C_2, \ldots, C_k\}$ adalah himpunan klaster dengan urutan ukuran $|C_1| \geq |C_2| \geq \ldots \geq |C_k|$. Didefinisikan *LC* dan *SC* sebagaimana Definisi 2. Untuk sembarang *record t*, didefinisikan *cluster-based local outlier factor* sebagaimana Persamaan (3).

$$CBLOF(t) = \begin{cases} |C_i| * \max(sim(C_j, t)) & \text{untuk } t \in C_i, C_i \in SC \text{ dan } C_j \in LC \\ |C_i| * (sim(C_i, t)) & \text{untuk } t \in C_i, C_i \in LC \end{cases} \quad (3)$$

Fungsi *sim*(*C*,*t*) pada Persamaan (3) adalah fungsi kemiripan transaksi *t* terhadap kelas *C* sebagaimana dalam algoritma Squeezer [8]. Meskipun *CBLOF* diperuntukkan untuk data kategorikal dan dapat dikembangkan untuk data numerik. Ini dilakukan dengan mendefinisikan *CBLOF* dengan perhitungan derajat *outlier* sebagaimana Persamaan (3).

### *NCBLOF* (Implementasi *CBLOF* pada Data Numerik)

Salah satu pendekatan deteksi *outlier* berbasis klaster adalah dengan mengesampingkan klaster-klaster kecil yang jauh dari klaster yang lain. Pendekatan ini dapat digunakan dengan menggunakan sembarang teknik klasterisasi, namun memerlukan *threshold* berapa jumlah minimum ukuran klaster dan jarak antara klaster kecil dengan klaster yang lebih besar. Pendekatan lain adalah dengan menentukan derajat dimana sebuah obyek berada pada sembarang klaster. Sebagai perwakilan klaster dapat digunakan *centroid* untuk menghitung jarak antara obyek dengan klaster.

Ada beberapa cara untuk mengukur jarak sebuah obyek ke sebuah klaster, yaitu dengan mengukur jarak sebuah obyek terhadap *centroid* terdekat, atau dapat juga dengan mengukur jarak relatif obyek dengan *centroid* terdekat. Jarak relatif adalah rasio jarak obyek terhadap *centroid* dibagi dengan jarak rata-rata semua titik terhadap *centroid* klaster di mana ia berada.

Hasil derajat *outlier* dapat dilihat berdasarkan *shading*. Pendekatan hanya berdasarkan jarak saja akan menyebabkan masalah jika set data mempunyai kerapatan yang berbeda-beda. Pada Gambar 2, dengan menggunakan jarak saja, obyek *D* tidak dianggap sebagai *outlier*, padahal obyek tersebut cenderung sebagai *outlier* lokal dari klaster terdekatnya. Sedangkan pendekatan pada Gambar 3, akan mengidentifikasikan *A*, *C*, dan *D* sebagai *outlier* sebagaimana Algoritma LOF [9].

Namun demikian jika sebuah obyek berada dalam klaster yang kecil, maka untuk perhitungan dengan jarak relatif seperti di atas ia tidak akan terdeteksi sebagai *outlier*. Oleh karena itu, pada penelitian ini digunakan pendekatan sebagaimana pada *CBLOF* yang menganggap obyek-obyek dalam klaster yang kecil sebagai kandidat *outlier*. Deteksi *outlier* menggunakan konsep mengenai klaster besar dan klaster kecil juga, dimana derajat *outlier* dihitung sebagai *Numerical Cluster-based Local Outlier Factor* (*NCBLOF*).

Dalam *CBLOF* ada dua komponen pembentukan derajat *outlier*, yaitu jumlah anggota klaster besar terdekat dan kemiripannya terhadap klaster terdekat tersebut. Dua komponen ini digunakan juga untuk mendefinisikan *NCBLOF* sebagaimana Persamaan (4).

$$NCBLOF(t) = \begin{cases} |C_j| \dfrac{1}{\text{relatif distance}(t, C_j)} & \text{untuk } t \in C_i, C_i \in SC \text{ dan } C_j \in LC, \\ & C_j = \arg\min(t, centroid(C_j)) \\ |C_i| \dfrac{1}{\text{relatif distance}(t, C_i))} & \text{untuk } t \in C_i, C_i \in LC \end{cases} \quad (4)$$

Rumus *NCBLOF* pada Persamaan (4) didefinisikan dengan menyesuaikan interprestasi derajat *outlier* pada *CBLOF* pada Persamaan (3).

## *MULTI CRITERIA DECISION MAKING* (*MCDM*)

*MCDM* adalah cabang dari masalah pengambilan keputusan, yang berkaitan dengan pengambilan keputusan, di bawah keberadaan sejumlah kriteria keputusan. Metode ini dibagi menjadi *Multi-objective Decision making* (*MODM)* dan *Multi-attribute decision making* (*MADM*). Metodologi ini mencakup adanya konflik antar kriteria, *incomparable unts*, dan kesulitan dalam pemilihan alternatif. Dalam *MODM*, alternatif-alternatif solusi tidak

ditentukan lebih dahulu. Melainkan sekumpulan fungsi obyektif dioptimasi terhadap sekumpulan konstrain atau batasan.

Dalam *MADM*, alternatif dievaluasi dengan mengatasi sekumpulan kriteria atau atribut yang saling konflik. Masalah penggabungan *outlier* dalam permasalahan penelitian ini termasuk dalam kategori ini. Masing-masing sub data numerik dan kategorikal dianggap sebagai sebuat atribut dalam *MADM*. Teori yang banyak digunakan dalam *MADM* adalah *multi-atribut value theory* (*MAVT*), dimana perangkingan alternatif keputusan dibangkitkan. Dalam prakteknya, metode berbasis *MAVT* menggunakan operator agregasi yang dirasa cocok untuk mendapatkan faktor *outlier* dari seluruh obyek. Operator tersebut di antaranya adalah operator *product* (kali), sum (tambah), dan operator $S_\infty$.

Berikut adalah macam-macam operator agregat yang dapat digunakan dalam *MAVT*:

1. Operator Perkalian $\prod$
   Operator perkalian juga dikenal sebagai metode perkalian berbobot. Operator ini menggunakan perkalian untuk menghubungkan rating dari atribut sebagai berikut:
   $\oplus( a_1, a_2, ..., a_m) = \prod (a_1^{w1}, a_2^{w2}, ..., a_m^{wm}) = a_1^{w1} a_2^{w2} ... a_m^{wm} = \prod a_i^{wi}$
2. Operator Penjumlahan
   Operator penjumlahan juga disebut dengan metode penjumlahan berbobot. Operator ini menggunakan penambahan untuk menghubungkan rating dari atribut sebagai berikut:
   $\oplus( a_1, a_2, ..., a_m) = +(w_1 a_1, w_2 a_2, ..., w_m a_m) = w_1 a_1 + w_2 a_2, ... + w_m a_m = \Sigma w_i a_i$
3. Operator $S_\infty$
   Operator $S_\infty$ juga dikenal dengan operator maksimum atau operator agregasi dasar. Operator ini memberikan nilai terbesar dari sekumpulan nilai yang diberikan sebagai berikut:
   $\oplus( a_1, a_2, ..., a_m) = S_\infty (w_1 a_1, w_2 a_2, ..., w_m a_m) = max \{ w_i a_i\}$

## ALGORITMA *MIXCBLOF*

Penelitian ini mengusulkan metode *MixCBLOF* untuk menyelesaikan masalah deteksi *outlier* pada set data campuran. Misalkan diberikan sebuah set data *D* yang terdiri dari *n* obyek dengan atribut campuran numerik dan kategorikal. Langkah-langkah Algoritma *MixCBLOF* adalah sebagai berikut:

1. Bagi set data campuran menjadi dua bagian, yaitu set data numerik, $D_1$ dan set data kategorikal, $D_2$.
2. Lakukan klasterisasi pada subset data numerik $D_1$ sehingga diperoleh sejumlah klaster $C_{11}$, $C_{12}$, ..., $C_{1p}$ dengan ukuran berturut-turut:
   $$|C_{11}| \geq |C_{12}| \geq ... \geq |C_{1p}|$$
   Tentukan klaster besar (*LC*) dan klaster kecil (*SC*) menggunakan Definisi 2.
3. Terapkan deteksi *outlier* berbasis klaster menggunakan atribut numerik, *NCBLOF*, terhadap obyek-obyek dalam klaster pada langkah 2 sebagaimana Persamaan (4).
4. Terapkan deteksi *outlier* berbasis klaster menggunakan atribut kategorikal terhadap obyek-obyek dalam klaster pada langkah 2 dengan *CBLOF* pada Persamaan (3).
5. Lakukan klasterisasi pada sub set data kategorikal sehingga diperoleh sejumlah klaster $C_{21}$, $C_{22}$, ..., $C_{2q}$ dengan ukuran berturut-turut:
   $$|C_{21}| \geq |C_{22}| \geq ... \geq |C_{2q}|$$
   Tentukan klaster besar (*LC*) dan klaster kecil (*SC*) menggunakan Definisi 2.
6. Terapkan deteksi *outlier* berbasis klaster menggunakan atribut kategorikal terhadap obyek-obyek dalam klaster pada langkah 2 dengan *CBLOF* pada Persamaan (3).
7. Terapkan deteksi *outlier* berbasis klaster menggunakan atribut numerik terhadap obyek-obyek dalam klaster pada langkah 5 dengan NCBLOF pada Persamaan (4).
8. Susun derajat *outlier* pada langkah 3, 4, 6, dan 7 dalam matrik keputusan $A=[a_{ij}]$.
9. Lakukan pembobotan secara *default* (bobot sama) atau dengan metode *Entropy*.
10. Gabungkan bobot *outlier* tiap obyek $t_1$, $t_2$, .., $t_n$ pada langkah 9 dengan fungsi agregat untuk mendapatkan derajat *outlier* akhir OF dari sebuah obyek $t_i$.
    $OF(t_i) = \oplus (a_{1i}, a_{2i}, a_{3i}, a_{4i})$

## HASIL DAN PEMBAHASAN

Algoritma *MixCBLOF* diimplemetasikan pada beberapa set data nyata yang diperoleh dari *UCI Machine Learning Repository* dengan

beberapa karakteristik khusus. Set data uji coba terdiri dari atribut campuran numerik dan kategorikal serta memiliki beberapa kelas atau klaster dimana sebagian di antaranya adalah kelas dengan ukuran yang relatif lebih kecil sehingga dapat dianggap sebagai sekumpulan *outlier*. Data yang digunakan pada uji coba ini adalah Set data *Cleveland* (*Heart Disease*), *Hypothyroid*, *Hepatitis*, dan *Annealing*. Dalam algoritma *MixCBLOF* ini melibatkan Algoritma *Squeezer* dan *CBLOF* untuk sub data kategorikal, sedangkan untuk data numerik digunakan Algoritma *CLUTO* [10] dan *NCBLOF*.

Uji coba dijalankan sesuai dengan skenario yang telah dirancang, yaitu:

1. Menentukan parameter yang tepat utuk Algoritma *MixCBLOF* meliputi penentuan $\alpha$, $\beta$, operator agregat, dan pembobotan yang tepat untuk masing-masing set data
2. Membandingkan *MixCBLOF* dibandingkan dengan algoritma lain, yaitu algoritma *CBLOF* yang diterapkan pada set data yang sudah didiskritisasi.

Evaluasi dilakukan dengan menggunakan *top ratio* dan *coverage*. *Top ratio* adalah perbandingan antara jumlah *k outlier* yang dihasilkan oleh algoritma (*n top ratio*) dengan jumlah keseluruhan obyek dalam data. Sedangkan *coverage* adalah rasio antara jumlah *outlier* eksak yang terdeteksi dengan jumlah keseluruhan *outlier* eksak yang dicari. Agar lebih mudah dalam melakukan analisa hasil, evaluasi dilakukan dengan melihat rata-rata pencapaian *coverage* untuk *top ratio* antara jumlah *outlier* eksak dengan jumlah keseluruhan data.

Hasil uji coba algoritma *MixCBLOF* dapat dilihat pada Tabel 1. Pencapaian *coverage* terbaik untuk *top ratio* antara jumlah *outlier* eksak dengan jumlah keseluruhan data dicetak dengan huruf tebal. Jika dilakukan rata-rata, algoritma *MixCBLOF* mencapai *coverage* 73.54%. Dari Tabel 1 dapat dilihat bahwa di antara operator yang ada, operator perkalian menghasilkan kinerja yang lebih baik jika dibandingkan dengan dua operator lainnya, yaitu penjumlahan dan maksimum. Selain itu, pembobotan sama menghasilkan kinerja yang lebih baik jika dibandingkan pembobotan dengan pembobotan berdasaran *entropy*.

Salah satu parameter yang juga penting, selain operator *agregat* dan pembobotan, adalah $\alpha$ dan $\beta$ yang mempengaruhi dipenuhinya konsep klaster besar dan kecil. Pada Tabel 2 dapat dilihat hasil pencapaian *coverage* untuk dua kasus, yaitu dipenuhinya konsep klaster besar dan kecil atau tidak. Berdasarkan hasil Tabel 2, tidak ada perbedaan yang signifikan terhadap dipenuhinya konsep klaster besar dan kecil. Namun demikian konsep ini tetap dibutuhkan berdasarkan definisi *CBLOF* yang dijelaskan di awal.

Tabel 1. Pencapaian *Coverage* untuk *n* = Jumlah *Outlier* Eksak pada Keseluruhan Set Data Berdasarkan Operator dan Pembobotan.

| Set data | *Coverage* (+) | | ∏ | | S∞ | |
|---|---|---|---|---|---|---|
| | *wi*=1 | *entropy* | *wi*=1 | *entropy* | *wi*=1 | *entropy* |
| Sub data *Cleveland* I | 76.90% | 53.80% | **92.30%** | 76.90% | 53.80% | 23.10% |
| Sub data *Cleveland* II | 77.00% | 77.00% | **89.00%** | 86.00% | 66.00% | 66.00% |
| *Dataset Cleveland* | 73.00% | 74.00% | **76.00%** | 75.00% | 69.00% | 68.00% |
| *Hypothyroid* | 72.10% | **73.00%** | 47.50% | 63.90% | 9.00% | 54.90% |
| *Hepatitis* | 52.40% | 33.30% | **66.70%** | 47.60% | 33.30% | 19.00% |
| *Annealing* | 35.30% | 32.40% | **47.10%** | 47.10% | 26.50% | 26.50% |
| Rata-rata | 64.45% | 57.25% | 69.77% | 66.08% | 42.93% | 42.92% |

Tabel 2. Perbandingan Kinerja *MixCBLOF* Dilihat dari Pemenuhan Konsep Klaster Besar dan Kecil.

| Set data | Dipenuhi Konsep Klaster Besar/Kecil | |
|---|---|---|
| | Iya | Tidak |
| *Sub Cleveland* I | 61.50% | 53.80% |
| *Hypothyroid* | 67.20% | 72.10% |
| *Hepatitis* | 66.70% | 66.7% |

| | | |
|---|---|---|
| *Annealing* | 47.10% | 47.10% |
| Rata-rata *Coverage* | 60.63% | 59.93% |

Tabel 3. Perbandingan Pencapaian *Coverage* Terbaik untuk *Top Ratio,* N=Jumlah *Outlier* Eksak, Antara *Mixcblof* dengan *CBLOF* Berbasis Diskritisasi Set Data.

| Set data | *Best Coverage* | | |
|---|---|---|---|
| | | *CBLOF* | |
| | *MixCBLOF* | *K-Means* | *Equal Width* |
| Sub data *Cleveland* I | 92.30% | 84.60% | 92.30% |
| Sub data *Cleveland* II | 88.60% | 82.90% | 88.60% |
| *Hypothyroid* | 73.00% | 66.40% | 16.40% |
| *Hepatitis* | 66.70% | 61.90% | 61.90% |
| *Annealing* | 47.10% | 44.10% | 38.20% |
| Rata-rata | 73.54% | 67.98% | 59.48% |

Tabel 4. *Running Time* Algoritma MixCBLOF dilihat dari Jumlah Atribut dan *Record.*

| Set data | Jumlah *record* | Jumlah atribut | *Running time* (detik) |
|---|---|---|---|
| Sub data *Cleveland* I | 177 | 14 | 0.125 |
| Sub data *Cleveland* II | 199 | 14 | 0.1563 |
| *Hypothyroid* | 2000 | 17 | 0.4688 |
| *Hepatitis* | 118 | 16 | 0.1406 |
| *Annealing* | 535 | 6 | 0.1406 |

Pada uji coba juga dilakukan perbandingan *MixCBLOF* terhadap Algoritma *CBLOF* dengan diskritisasi atribut numerik. Hasil dari keseluruhan uji coba dirangkum pada Tabel 3. Dari Tabel 3 dapat dilihat bahwa Algoritma *MixCBLOF* dapat menyelesaikan masalah deteksi *outlier* pada set data campuran dengan cukup baik. *Top ratio* antara jumlah *outlier* eksak dengan jumlah keseluruhan data mampu mencapai rata-rata *coverage* sebesar 73.54 %. Hasil ini lebih baik jika dibandingkan *CBLOF* dengan diskritisasi numerik yang hanya mampu mencapai rata-rata *coverage* 67.98% untuk diskritisasi dengan *K-Means* dan 59.48% untuk diskritisasi dengan *equal width*.

Tabel 4 menampilkan informasi mengenai *running time* dari algoritma *MixCBLOF* jika dilihat dari jumlah atribut dan *record* pada masing-masing kasus. Uji coba ini dilakukan pada lingkungan sebagai berikut.

1. *Hardware*:
   a. *Processor: Dual Core Genuine Intel (R) CPU T2080 @* 1.73 *GHz*
   b. *Memory: DDR* 512 *MB*
   c. *Hard disk*: 80 *GB*
2. *Software*:
   a. *Microsoft Windows XP Professional Version 2002 Service Pack 2*
   b. *MATLAB Versi* 7.0

## SIMPULAN DAN SARAN

Dari uji coba dan pembahasan yang telah dilakukan dapat ditarik simpulan sebagai berikut:

1. Berkaitan dengan penggunaan parameter Algoritma *MixCBLOF*:
   a. Penetapan nilai $\alpha$ dan $\beta$ yang tepat diperlukan untuk mendapatkan konsep klaster besar dan kecil. Hal ini berguna untuk mendapatkan definisi *outlier* yang sesungguhnya sesuai dengan konsep *outlier* berbasis klaster. Mengenai besaran nilai $\alpha$ dan $\beta$ dapat ditentukan dengan melihat hasil klasterisasi pada kedua *subset* data numerik dan kategorikal.
   b. Operator perkalian menghasilkan rata-rata *coverage* lebih baik dibandingkan dua operator penjumlahan dan maksimum untuk *top ratio* sejumlah *outlier* eksak.
   c. Pembobotan sama ($w_i$=1) menghasilkan rata-rata *coverage* lebih baik daripada pembobotan berdasarkan *entropy* untuk *top ratio* sejumlah *outlier* eksak.
2. Algoritma *MixCBLOF* dapat menyelesaikan masalah deteksi *outlier* pada set data dengan atribut campuran dengan baik, yaitu dengan rata-rata *coverage* 73.54%. Hasil ini lebih baik daripada menggunakan metode diskritisasi atribut numerik yang hanya mencapai *coverage* 67.98% dengan menggunakan metode *unsupervised* seperti *K-Means* dan 59.48% dengan menggunakan *equal width*.

Untuk pengembangan lebih lanjut, dapat dilakukan dengan mencari sebuah metode yang dapat secara otomatis menentukan parameter *treshold s* dan *k* pada metode klasterisasi. Ide yang dapat digunakan adalah dengan mengoptimalkan hasil klasterisasi yang mampu menghasilkan klaster-klaster besar dan kecil dengan ukuran yang jauh berbeda.

## DAFTAR PUSTAKA

[1] Tan PN, Steinbach M, and Kumar V. *Introduction to Data Mining*. Boston: Pearson Addison Weisley. 2006.

[2] He Z, Deng X, and Xu X. *Clustering Mixed Numeric and Categorical Data*: *A Cluster Ensemble Approach*. eprint arXiv:cs/0509011. 2005. URL: http://arxiv.org/ftp/cs/papers/0509/0509011.pdf, diakses tanggal 20 November 2009.

[3] Assent I, Krieger R, Muller E, and Seidl T. Subspace Outlier Mining in Large Multimedia Databases. *Dagstuhl Seminar Proceedings 07181: Parallel Universes and Local Patterns*. 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.4039&rep=rep1&type=pdf, diakses tanggal 20 Nopember 2009.

[4] Aggarwal C and Yu P. An Effective and Efficient Algorithm for High-dimensional Outlier Detection. *VLDB Journal.* 14: 211-221. 2005.

[5] Hong Y, Kwong S, Chang Y and Ren Q. Unsupervised Data Pruning for Clustering of Noisy Data. *Elvesier: Knowledge-Based System*. 21: 612-616. 2008.

[6] He Z, Xu X and Deng S. Discovering Cluster-based Local Outliers. *Pattern Recognition Letter.* 24: 1641-1650. 2003.

[7] Climaco J. *Multicriteria analysis*. New York: Springer-Verlag. 1997.

[8] He Z, Xu X and Deng S. Squeezer: An Efficient Algorithm for Clustering Categorical Data. *Journal of Computer Science and Technology*. 17: 611-624. 2002.

[9] Breunig M M. Kriegel. HP, Ng RT and Sander J. LOF: Identifying Density-based Local Outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 93-104. 2000.

[10] Anonim. *CLUTO 2.1.1. Software for Clustering High-Dimensional Dataset*. URL: http://www.cs.umn.edu/~karpys, diakses tanggal 20 November 2009.