J. Math. and Its Appl.
ISSN: 1829-605X
Vol. 2, No. 2, Nov 2005, 117–129

Kajian Algoritma GDBScan, Clarans dan Cure untuk Spatial Clustering

Budi Setiyono, Imam Mukhlash
Jurusan Matematika FMIPA ITS
Kampus ITS Sukolilo Surabaya, 60111
mazh_budi@yahoo.com, immukhlash_its@yahoo.com

Abstrak

Abstrak Spatial data mining merupakan salah satu bidang kajian dalam data mining dan menjadi salah satu bidang yang sangat cepat perkembangannya. Salah satu cabang dari spatial data mining adalah geographic data mining. Geographic data mining adalah penemuan pengetahuan baru dari sejumlah besar data geo-spatial (geo-reference). Beberapa metode dalam data mining telah dikembangkan para ahli. Salah satu metode yang paling banyak dikembangkan adalah clustering. Pada penelitian ini akan dilakukan kajian tentang tiga buah algoritma, yaitu algoritma density-based clustering, algoritma CLARANS clustering, serta algoritma CURE. Selanjutnyadilakuan implementasi dalam bentuk perangkat lunak. Studi kasus yang digunakan adalah clustering wilayah (peta) kota Surabaya berdasarkan parameter rasio jumlah penduduk miskin dan sangat miskin, kepadatan, dan tingkat kesejateraan tiap-tiap kelurahan kota Surabaya.

Kata kunci: *Geographic data mining, density-based clustering, CLARANS, CURE*

1. Pendahuluan

Mengikuti definisi dari data mining, spatial data mining adalah penemuan pengetahuan dari sejumlah besar data spasial. Salah satu cabang dari spatial data mining adalah geographic data mining (GDM). Salah satu pendukung utamanya adalah sistem basisdata spasial (Spatial Database Systems, SDBS). Beberapa metode yang dikembangkan dalam geographic data mining merupakan kasus khusus dari metode dalam spatial data mining. Metode-metode dalam spatial data mining yang dikembangkan secara garis besar dapat dikelompokkan kedalam beberapa macam, antara lain generalisasi, klasifikasi, clustering, prediksi, outlier analysis, dan deteksi trend [3,6,8]. Clustering merupakan metode yang paling banyak dikembangkan. Clustering adalah identifikasi dan pengelompokan dari class (yang disebut cluster) untuk suatu himpunan obyek sedemikian hingga anggota dari suatu cluster mempunyai sifat yang mirip dengan sesama anggota cluster. Algoritma-algoritma spatial clustering secara garis besar dikelompokkan menjadi tiga tipe yaitu patitioning-based clustering, density-based clustering dan hierarchical clustering [2]. Ruang lingkup dari artikel ini akan dipapaparkan dua aspek yaitu pengembangan perangkat lunak GIS dan analisis secara asimptotis dan empiris dari algoritma GDBSCAN, CLARANS, serta CURE untuk geografik data mining. Sumber data yang digunakan adalah data spasial dan non-spasial. Sumber data spasial yang digunakan adalah peta administratif Kota Surabaya tahun 2000 yang dikembangkan oleh LPPM-ITS, sedangkan data non-spasial yang digunakan adalah data pendapatan perkapita, jumlah penduduk miskin dan jumlah penduduk sangat miskin di Kota Surabaya yang didapatkan dari BPPS Kota Surabaya. Selanjutnya, dengan ditemukannya cluster secara geografis berdasarkan tingkat kesejahteraan penduduknya, maka kita bisa mengetahui profil kesejahteraan penduduk berdasarkan wilayah dan hasilnya dapat lebih dimanfaatkan lebih jauh dan lebih efektif untuk dasar pengambilan keputusan.

2. Spatial Clustering

Sebagaimana spatial data mining, spatial clustering merupakan salah satu bagian dari clustering secara umum. Perbedaan utama antara spatial clustering dan clustering secara umum terletak pada kenyataan bahwa proses spatial tidak selalu murni terjadi secara random [1]. Untuk merepresentasikan data yang akan dicluster biasanya digunakan dua bentuk struktur data yaitu matrik data dan matrik ketidaksamaan. Matrik data merepresentasikan n obyek (pada kasus ini misalnya obyek spasial) dengan p variabel (disebut juga pengukuran atau atribut), yang

kemudian dinyatakan dengan matrik berukuran $n \times p$

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{1}$$

sedangkan matrik ketidaksamaan merepresentasikan himpunan 'kedekatan' atau 'jarak' antara satu obyek ke obyek yang lain

$$\begin{pmatrix} d(1,1) & d(1,2) & \cdots & d(1,n) \\ d(2,1) & d(2,2) & \cdots & d(2,n) \\ \vdots & \vdots & \vdots & \vdots \\ d(n,1) & d(n,2) & \cdots & d(n,n) \end{pmatrix} \tag{2}$$

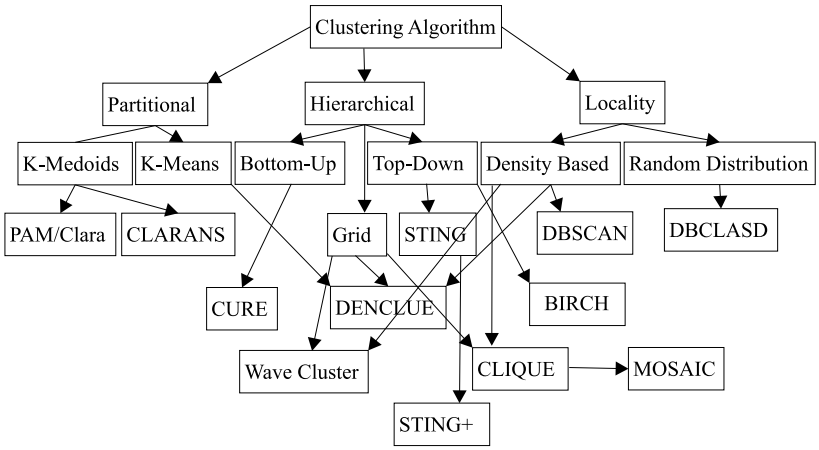
yang mana $d(i,j)$ menyatakan 'kedekatan' atau 'jarak' dari obyek i dan j. Karena jarak antara suatu obyek dengan dirinya sendiri adalah 0 ($d(i,i) = 0$) dan $d(i,j) = d(j,i)$ maka matrik diatas dapat dinyatakan dalam bentuk matrik segitiga atas maupun matrik segitiga bawah. Pada beberapa aplikasi tertentu, sebelum dilakukan penghitungan jarak antar obyek kadang-kadang perlu dilakukan standardisasi. Proses standardisasi dapat dilakukan dengan dua langkah yaitu menghitung defiasi absolute mean kemudian menghitung ukuran yang terstandardisasi (z-score). Untuk menghitung jarak antara satu obyek dengan obyek yang lain dapat digunakan tiga perhitungan yaitu dengan jarak Euclid, jarak Manhattan dan jarak Minkowski [5,7]. Mencari kesamaan merupakan pekerjaan utama dalam proses clustering. Sebuah cluster c_j adalah sebuah subset dari n obyek yaitu $c_j \subseteq X$. n obyek dipartisi menjadi k cluster yang berbeda $C = \{c_1, c_2, \cdots, c_k\}$ sedemikian hingga

$$c_i \cap c_j = \varnothing, \quad \forall i, j \in \{1, 2, \cdots, k\}, i \neq j \tag{3}$$

dan

$$c_1 \cup c_2 \cup \cdots \cup c_k = X$$

dengan $c_j \in R^n$ dan $j = 1, 2, \cdot, k$ didasarkan pada ukuran kesamaan. Secara umum, proses dalam algoritma clustering berisi langkah-langkah untuk mendapatkan $c_1, c_2, \cdots, c_k$. Pavel Berkhin mengklasifikasikan algoritma clustering menjadi beberapa macam antara lain *hierarchical methods, partitioning methods, grid-based methods, methods based on co occurence of categorical data, constraint based clustering, scalable clustering algorithm* dan *clustering algorithm for high dimensional data*[8]. Kolatch mengklasifikasikan algoritma spatial clustering menjadi tiga macam yaitu partitional clustering, locality-base clustering dan hierarchical clustering. Hubungan dari berbagai macam metode tersebut dan algoritma yang terdapat di dalam masing-masing metode dapat digambarkan pada Gambar 1.



Gambar 1: Klasifikasi Algoritma Clustering

2.1. Algoritma GDBSCAN

Algoritma ini merupakan salah satu algoritma clustering berbasis kerapatan (density-based clustering). GDBSCAN digunakan untuk menggali kumpulan dari density-connected dari obyek tetangga yang mempunyai kesamaan nilai dari atribut non-spatial. Secara umum, algoritma GDBSCAN adalah algoritma yang digunakan untuk proses clustering obyek-obyek spasial yang didasarkan pada atribut spasial dan non-spasial[4].

Proses untuk menemukan cluster dilakukan dengan cara pengujian terkait dengan parameter ϵ-neighborhood (ketetanggaan dalam radius ϵ) dari setiap titik yangada dalam database. Jika ϵ-neighborhood dari p berisi lebih dari MinPts maka dapat ditemukan cluster baru dengan p sebagai core object (ϵ-neighborhood memuat minimal sama dengan MinPts). Proses ini akan dilakukan secara berulang sampai tidak ada lagi obyek yang dapat ditambahkan pada cluster yang ada. Dengan demikian, inti dari algoritma ini adalah menentukan core object-nya kemudian melakukan ekspansi cluster.

Langkah-langkah dalam algoritma GDBSCAN untuk membangun cluster adalah sebagai berikut

1. Inisialisasi ClusterId, ClusterId = 1.

2. Lakukan langkah 3 sampai 5 untuk setiap obyek yang akan diproses.

3. Ambil O sebagai obyek ke i, dimana $i = 1, 2, 3, \cdots N$ dengan N adalah jumlah obyek yang akan diproses. Jika $i = N$ maka proses berhenti.

4. Jika status O bernilai UNCLASSIFIED, lakukan langkah 5 dan jika salah maka kembali ke langkah 3.

5. Jika Fungsi ExpandCluster bernilai benar, maka dilakukan pembentukan cluster baru dengan memberikan nilai pada ClusterId menjadi $ClusterId = ClusterId + 1$ dan jika bernilai salah kembali ke langkah 3.

ExpandCluster digunakan untuk membangun cluster dan memperluas wilayah cluster yang memenuhi parameter ϵ dan MinPts. Fungsi ExpandCluster terhadap suatu obyek O memiliki langkah-langkah sebagai berikut:

1. Ambil himpunan seeds adalah obyek O dan obyek neighborhood dari obyek O yang memenuhi ϵ.

2. Jika himpunan seeds memenuhi kondisi $wCard(seeds) < MinPts$, dimana wCard adalah fungsi penghitung jumlah obyek, maka obyek O dikategorikan sebagai NOISE dan ExpandCluster akan memberikan nilai balikan False. Jika himpunan seeds tidak memenuhi kondisi tersebut, maka lanjutkan ke langkah 3.

3. Masukkan seeds sebagai anggota dari cluster C(clusterId).

4. Hapus obyek O dari seeds.

5. Ambil P adalah anggota himpunan seeds yang pertama.

6. Ambil himpunan result adalah obyek P dan obyek "tetangga" dari obyek P yang memenuhi ϵ.

7. Jika himpunan result memenuhi kondisi wCard (result) = MinPts, maka lakukan langkah 8 sampai 11.

8. Ambil Q adalah obyek ke i, dimana $i = 1, 2, 3, \cdots, M$, dan M adalah jumlah dari himpunan result.

9. Jika $wCard(Q) > 0$ dan status Q bernilai UNCLUSSIFIED, maka tambahkan Q ke dalam anggota himpunan seeds.

10. Masukkan Q sebagai anggota seeds.

11. Masukkan Q sebagai anggota C(clusterId).

12. Hapus Obyek P dari seeds.

13. Ulangi langkah 5 sampai 12 sampai anggota himpunan seeds habis.

2.2. Algoritma Clarans

CLARANS adalah algoritma k-medoid. Algoritma ini melanjutkan kerja dari Algoritma PAM dan CLARA dengan melakukan pencarian graph secara acak untuk mendapatkan medoid-medoid yang mewakili sejumlah cluster[9]. Medoid adalah data point yang terletak pada tengah-tengah Group. Algoritma ini menggunakan maxneighbour dan numlocal sebagai parameter. Maxneighbbour adalah nilai maximum dari node sekawan yang diuji. Numlocal adalah nilai maksimal dari local minimum yang dapat dikumpulkan. Secara umum, langkah-langkah dalam algoritma CLARANS adalah:

1. Masukkan parameter Numlocal dan maxneighbor. Inisialisasi i menjadi 1, dan mincost bilangan yang besar.
2. Set current menjadi node yang acak pada $G_{n,k}$
3. Set j menjadi 1.
4. Pilih obyek tetangga secara acak S dari obyek current, dan dengan persamaan yang ada, hitung perbedaan biaya dari kedua node.
5. Jika S mempunyai biaya yang lebih rendah, ganti current dengan S, dan kembali ke langkah 3.
6. Jika tidak, naikkan j. Jika $j = maxneighbor$, kembali ke Langkah 4.
7. Selanjutnya, jika $j > maxneighbor$, bandingkan biaya dari current dengan mincost. Jika biaya current lebih kecil dari mincost, update mincost dengan biaya current dan ganti bestnode dengan current.
8. Naikkan i satu nilai. Jika $i > numlocal$, hasilkan bestnode dan hentikan proses. Dan jika tidak, kembali ke Langkah 2.

Langkah 3 sampai 6 di atas melakukan pencarian node dengan biaya yang lebih rendah. Tetapi, jika node current dibandingkan dengan jumlah maksimum dari node sekawan (maxneighbor) dan masih memberikan biaya yang paling rendah, node current dinyatakan menjadi lokal minimum. Kemudian, pada langkah 7, biaya dari lokal minimum ini dibandingkan dengan biaya paling rendah yang diperoleh sejauh ini. Yang paling rendah dari kedua biaya di atas disimpan dalam mincost. Algoritma CLARANS kemudian mengulangi untuk mencari lokal minima yang lain, sampai nilai numlocal dipenuhi.

Fungsi yang digunakan pada langkah [5] diatas diambil dari PAM. Persamaan ini digunakan untuk menghitung C_{ih} yaitu biaya pergantian (swap) antara current medoid (Oi) dengan non-medoid (O_i). Notasi umum Cjih adalah :

$$C_{jih} = d(O_j, O_h) - d(O_j, O_i)$$

Sedangkan d(O1,O2) diperoleh dari :

$$d(O_1, O_2) = minOed(O_1, O_2)$$

dimana notasi minOe menandakan nilai minimum atas semua medoids Oe dan notasi $d(O_1; O_2)$ menyatakan jarak atau perbedaan antara obyek O_1 dan O_2.

2.3. Algoritma CURE (Clustering Using Representative)

Pada agglomerative (bottom-up) dikenal adanya algoritma CURE. Dengan algoritma CURE (Clustering Using Representatives) akan didapatkan cluster berupa pengelompokan wilayah-wilayah yang didasarkan pada data atribut spasial dan non-spasial [10]. Hasil cluster wilayah tersebut merupakan suatu item informasi yang dapat digunakan sebagai acuan untuk pengambilan keputusan terkait dengan data yang ada.

Secara garis besar, algoritma CURE menggunakan algoritma baru dalam clustering secara hirarki dengan mengadopsi suatu 'middle ground' antara pendekatan berbasis centroid dengan pendekatan berbasis obyek yang representativ (semua titik-titik ekstrim). Hal ini disebabkan oleh pemilihan pusat cluster (sementara) yang tidak satu atau semua obyek yang mungkin seperti dalam agglomerative clustering, tetapi menggunakan sejumlah tertentu cluster yang cukup mewakili ketersebaran (well-scattered). Pemilihan titik-titik yang representative berikutnya (level di atasnya) dilakukan dengan cara menyusutkan jumlah cluster menjadi lebih sedikit dengan suatu parameter yang dinamakan factor penyusutan (shrinking factor). Pada setiap tahap dalam algoritma, akan dilakukan penggabungan dua cluster - diwakili oleh dua titik representative- yang mempunyai kedekatan paling kecil.

Untuk menangani database yang besar, CURE menggunakan penggabungan antara sampling secara random dan pemartisian. Pertama kali sample random dipartisi, kemudian setiap partisi dicluster secara parsial. Cluster-cluster yang dihasilkan kemudian di-cluster lagi (tahap kedua) untuk menghasilkan cluster yang diinginkan. Untuk menghentikan proses clustering digunakan suatu parameter, misalkan k, yang berisi jumlah cluster yang dihasilkan. Algoritma ini mirip dengan proses generalisasi, sehingga sangat sesuai jika penggunaannya digabungkan dengan generalisasi.

Langkah-langkah pada Algoritma CURE dapat digambarkan sebagai berikut :

1. Gambarkan contoh obyek-obyek S
2. Bagi obyek-obyek S dalam partisi tertentu
3. Kelompokkan sebagian cluster pada masing-masing partisi

4. Eliminasi outlier dengan contoh acak (random sampling). Jika cluster terlalu jauh, eliminasi itu.

5. Kelompokkan sebagian cluster. Dengan mengunakan faktor penyusutan α.

6. Pelabelan data pada disk.

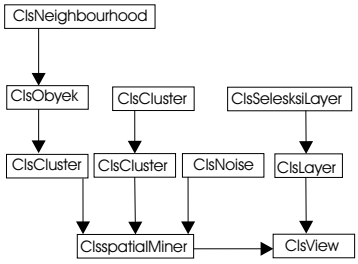
3. Pengembangan Perangkat Lunak

Secara garis besar, perangkat lunak diimplementasikan dalam bentuk obyek-obyek (class) beserta modul-modul yang terkait dengan clustering dan penyimpanan data menggunakan tabel-tabel. Class-class dan modul-modul yang ada diimplementasikan dengan Visual Basic dengan memanfaatkan obyek-obyek dalam MapObject 2.0 untuk akses obyek-obyek peta. DBMS yang digunakan adalah SQL Server dengan kemampuan untuk menyimpan data peta spasial dan nonspasial. Sistem yang dibangun terdiri dari atas empat class utama, yaitu class obyek, class neighborhood, class cluster, dan class noise (Gambar 2). Class obyek digunakan untuk menyimpan data spasial beserta atribut non-spasial, simbolisasi tematik berupa warna obyek pada peta, obyek "tetangga", dan status obyek yang dibagi dalam tiga kategori, yaitu CLUSTERED yang menandakan bahwa obyek termasuk dalam salah satu cluster yang terbentuk, NOISE yang menandakan bahwa obyek termasuk noise, dan UNCLUSSIFIED yang menandakan bahwa obyek belum dilakukan proses membangun cluster. Class neighborhood berguna untuk menyimpan obyek-obyek "tetangga" dari suatu obyek tertentu. Class cluster digunakan untuk menyimpan obyek-obyek yang memenuhi kriteria clustering, sedangkan class noise digunakan untuk menyimpan obyek yang tidak memenuhi kriteria. Rancangan basis data yang yang telah terintegrasi memuat atribut spasial dan non-spasial terdiri dari enam tabel, yaitu Tabel Layer Kecamatan, Tabel Layer Kelurahan, Tabel Kecamatan, Tabel Kelurahan, Tabel Sensus Kecamatan, Tabel Sensus Kelurahan. Hubungan antar table digambarkan dalam bentuk Entity Relationship Digram (ERD) yang ditampilkan pada Gambar 3.

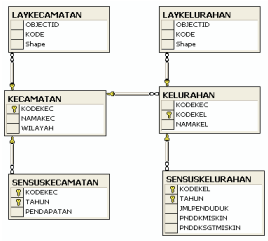
Proses persiapan data akan berhubungan langsung dengan basis data Microsoft SQL Server untuk menghasilkan obyek yang akan disimpan ke dalam class obyek.

4. Uji Coba Perangkat Lunak

Lingkungan Uji Coba Perangkat keras yang dipakai untuk uji coba perangkat lunak ini adalah komputer dengan Processor Intel Pentium IV 1700 MHz dengan memori utama 376 MB yang telah diinstall Microsoft SQL Server 2000 Enterprise



Gambar 2: Diagram Obyek



Gambar 3: ER-Diagram

Manager dengan Sistem Operasi Microsoft Windows XP Professional SP 2. Untuk melakukan pengujian Dataset yang dipersiapkan antara lain :

a. Tabel Kecamatan dan Tabel LayKecamatan: 28 record data

b. Tabel Kelurahan dan Tabel Laykelurahan: 163 record data

c. Tabel SensusKecamatan : 28 record data

d. Tabel SensusKelurahan : 163 record data

5. Perbandingan Hasil Uji Coba

5.1. Perbandingan kompleksitas

- GDBSCANS
  Untuk basis data spasial yang tidak menggunkan sistem indexing, kompleksitas pencarian neighborhood adalah $O(n)$, sedangkan basis data spasial yang menggunkan sistem indexing, kompleksitas pencarian neighborhood adalah $O(\log n)$. Pada makalah ini basis data spasial yang digunakan menggunakan sistem indexing, sehingga kompleksitas algoritma GDBSCAN adalah $O(n \log n)$.

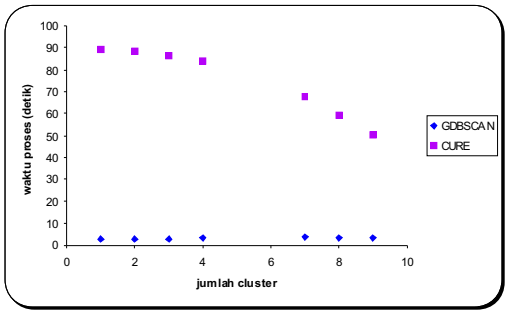
- CLARANS
  Jika jumlah maksimum tetangga diset menjadi $(k-1)$ maka hasil dari clustering yang diproduksi CLARANS sama dengan PAM. Karena kerja terbesar dari CLARANS adalah pada saat pengujian obyek. Sehingga kompleksitas CLARANS adalah $O(kn^2)$.

- CURE
  Kasus kompleksitas waktu terburuk dari algoritma clustering CURE dapat

ditunjukkan menjadi $O(n^2 \log n)$. Dapat ditunjukkan bahwa ketika dimensionalitas titik-titik data kecil, maka kompleksitas waktu jauh berkurang ke $O(n^2)$. Jika heap atau k-d tree memerlukan ruang linear, maka kompleksitas ruang dari algoritma kami adalah $O(n)$.

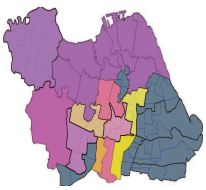
5.2. Perbandingan Hasil Clustering

Berikut ini diberikan grafik yang menyatakan hubungan antara jumlah cluster yang terbentuk dengan waktu proses (Gambar 4).

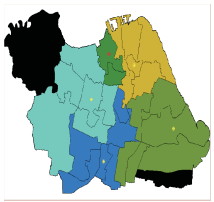


Gambar 4: Grafik Perbandingan CURE dan GDBSCAN pada Jumlah Cluster terhadap Waktu Proses

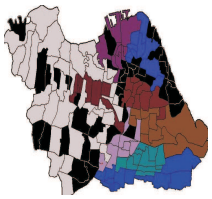
Dari hasil uji coba diatas didapatkan bahwa algoritma GDBSCAN mempunyai waktu proses yang lebih baik daripada algoritma CURE di dalam membentuk jumlah cluster yang sama, ini terlihat pada grafik pada Gambar 4. Dari grafik pada Gambar 4 perbandingan jumlah noise dengan jumlah cluster yang terbentuk sama, didapatkan jumlah noise pada algoritma CURE dan CLARANS lebih sedikit dibandingkan dengan algoritma GDBSCAN. Berikut adalah salah satu perbandingan cluster (Gambar 5, 6, dan 7) yang dihasilkan antara algoritma GDBSCANS, CLARANS dan CURE



Gambar 5: GDBSCANS



Gambar 6: CLARANS



Gambar 7: CURE

6. KESIMPULAN

Dari pembahasan diatas dapat diambil beberapa kesimpulan antara lain:

1. Penentuan obyek awal, parameter ketetanggaan dan nilai MinCard pada proses clustering berpengaruh terhadap cluster yang dihasilkan oleh algoritma GDBSCAN. Cluster-cluster yang dihasilkan kemungkinan mempunyai 'tingkat penyebaran' yang tinggi secara spasial, sehingga kadang kurang tampak sebagai sebuah cluster.

2. Dengan algoritma CURE, beberapa parameter yang mempengaruhi proses clustering adalah jumlah cluster yang diinginkan, predikat NPred spasial dan non-spasial, jumlah partisi awal dan jumlah partisi dari masing-masing partisi.

3. Pada penggunaan algoritma CLARANS, beberapa parameter yang mempengaruhi proses clustering adalah maxneighbour dan numlocal. Maxneighbour adalah nilai maximum dari node sekawan yang diuji, sedangkan numlocal adalah nilai maksimal dari local minimum yang dapat dikumpulkan. Selain itu, jumlah cluster yang diinginkan juga mempengaruhi proses clustering.

4. Secara umum, algoritma GDBSCAN mempunyai waktu proses yang lebih cepat, sedangkan algoritma CLARANS menhasilkan noise yang lebih sedikit dan cluster yang dihasilkan lebih baik daripada yang lain.

Pustaka

[1] Bin Jiang, Spatial Clustering for Mining Knowledge in Support of Generalization Processes in GIS ICA Workshop on Generalisation and Multiple representation, Leicester, 20-21 August 2004

[2] Erika Kolatch, Clustering Algorithm for Spatial Databases: A Survey, Department of Computer Science University of Maryland, 2001

[3] Harvey J. Miller, Geographic Data Mining and Knowledge Discovery, in J. P. Wilson and A. S. Fotheringham (eds.) Handbook of Geographic Information Science, 2004, in press.

[4] J. Sander, Ester, M., Kriegel, H.-P., and Xu X, Density-based Clustering in Spatial Databases: the Algorithm GDBSCAN and Its Applications, in Data Mining and Knowledge Discovery 2(2), Kluwer Academic Publisher, 1998

[5] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kauffmann Publisher, 2001

[6] Martin Ester, Alexander Frommelt, Hans Peter Kriegel, Jorg Sander, Spatial Datamining: Database Primitives, Algorithms and Efficient DBMS Support, Datamining and Knowledge Discovery, 4, 193-216, Kluwer Academic Publisher, 2000

[7] Pang Ning Tang, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Addison Wesley, 2005

[8] Pavel Berkhin, Survey of Clustering Data Mining Techniques, Accrue Software Inc., 2003

[9] Ng Raymond T., and Han J, "CLARANS: A Method for Clustering Objects for Spatial Data Mining ", IEEE Transaction on Knowledge and Data Engineering, vol 14, no.5, 2002

[10] Guha, S., Rastogi, R., Shim, K., "CURE: An Efficient Clustering Algoritm for Large Database". 1998