Implementasi *Density Based Clustering* menggunakan *Graphics Processing Unit* (GPU)

[1]Dede Nofrianda Utama, [2]Fhira Nhita, [3]Izzatul Ummah

Program Studi Ilmu Komputasi Universitas Telkom, Bandung

[1]dnofrianda@gmail.com , [2]finalproject.vir@gmail.com , [3]izzatul.ummah@gmail.com

Abstraksi

Data merupakan sumber informasi yang berguna untuk kelangsungan hidup manusia. Untuk menjadikan data tersebut bermanfaat, diperlukan suatu metode yang dapat menggali informasi penting dari data yang ada. Salah satu metode penarikan informasi dari sekumpulan data dikenal dengan Data Mining. Teknik menambang informasi pada Data Mining pun beragam, salah satunya Clustering. Clustering merupakan metode pengelompokkan data yang memiliki kesamaan atribut kedalam satu kelompok dengan aturan tertentu. Pada penelitian ini algoritma Clustering yang digunakan adalah Density Based Spatial Clustering Application with Noise (DBSCAN). DBSCAN merupakan algoritma Cluster yang bersifat density-based, yaitu mengelompokkan data berdasarkan kepadatannya ke dalam satu kelompok, dan data yang jarang pada kelompok lainnya. Untuk mengelompokkan data dengan dimensi yang tinggi, diperlukan perangkat yang dapat meminimalkan biaya komputasi. GPU (Graphics Processing Unit) memungkinan mengolah data dengan dimensi tinggi dalam waktu yang singkat. Jika GPU dikombinasikan dengan DBSCAN pengelompokkan data dapat menghasilkan performansi kerja algoritma yang baik dengan akurasi yang tinggi serta biaya komputasi yang minimum. Salah satu metode penerapan GPU pada DBSCAN dengan melakukan perhitungan jarak antar data secara paralel di GPU. Hasil perhitungan ini mampu menghemat biaya komputasi rata – rata sebesar 0,9734 detik untuk data dengan dimensi 15154 dan 0,063 detik untuk data dengan dimensi 12600. Selain itu pada evaluasi performansi, GPU menghasilkan nilai yang cukup baik dibandingkan dengan algoritma serialnya.

Kata kunci : Data Mining, Clustering, Density Based Spatial Clustering Application with Noise (DBSCAN) , density based, GPU (Graphics Processing Unit) .

Abstract

Data is a source of useful information for human survival. To make the data useful, we need a method that can dig up important information from existing data. One method of information retrieval of a set of data known as Data Mining. Mine engineering information on Data Mining also varied, one of them Clustering. Clustering is a method of grouping data that have similar attributes into one group with certain rules. In this research, Clustering algorithm used is Density Based Spatial Clustering Applications with Noise (DBSCAN). DBSCAN a Cluster algorithm that is density-based, ie classifying data based on the density into one group, and data are rare in other groups. To classify high-dimensional data, the device needs to minimize the costs of computing. GPU (Graphics Processing Unit) allows to process high-dimensional data in a short time. If the GPU is combined with DBSCAN grouping data can generate peformance algorithms work well with high accuracy and minimum computational cost. One of method applying the GPU on DBSCAN by calculating the distance between the data in parallel on the GPU. The result of this calculation is able to save the cost of computing with time 0.9734 sec for data with dimensions of 15154 and 0.063 seconds for the data to the dimensions of 12600. In addition to the performance evaluation, GPU produces a pretty good value compared with serial algorithm.

Keywords : Data Mining, Clustering, Density Based Spatial Clustering Application with Noise (DBSCAN) , density based, GPU (Graphics Processing Unit) .

1. Pendahuluan

Data berisikan informasi yang dapat diolah untuk menunjang kehidupan manusia. Data pada bidang kesehatan contohnya, dapat membantu seorang dokter menganalisa suatu gejala pada pasien untuk selanjutnya dapat dilakukan tindakan pengobatan yang tepat. Pengolahan terhadap suatu data inilah yang memungkinkan penarikan informasi dapat optimal dan dapat digunakan untuk suatu penelitan ataupun sistem pengambilan keputusan.

Data mining merupakan suatu metode analisis untuk mendapatkan informasi dengan menggali data dalam jumlah yang besar[1,4,9]. Metode yang terdapat dalam data mining diantaranya, classification, Clustering, dan association. Clustering bertujuan mengelompokkan suatu data yang memiliki kesamaan atribut. Semakin banyak atribut suatu data, semakin besar biaya komputasi yang diperlukan untuk mengelompokkan data tersebut[1].

Density Based Spatial Clustering Application with Noise (DBSCAN) adalah salah satu algoritma Clustering yang menggunakan kepadatan atribut data, untuk kemudian dikelompokkan dalam satu Cluster, dan atribut yang jarang ke dalam kelompok Cluster lainnya. Cluster berbasis kepadatan ini memiliki akurasi dalam penentuan kelompok lebih baik dibanding dengan algoritma Clustering lainnya, seperti partitioning atau Clustering hierarchy[9]. Selain itu, kelebihan algoritma ini mampu mengatasi noise dengan baik, dan mampu mengolah data dalam jumlah besar dalam waktu singkat[9].

GPU (Graphic Processing Unit) membantu meningkatkan pengolahan data secara paralel dengan konsumsi memori yang sedikit[1]. Dengan keuntungan tersebut GPU dapat dikombinasikan dengan algoritma Cluster berbasis kepadatan untuk mengolah data berdimensi tinggi, namun dengan biaya komputasi yang sedikit. Salah satu contoh penerapannya terdapat pada penelitian yang dilakukan oleh Christian Bohm, Robert Noll, Claudia Plant, dan Bianca Wackersreuther dengan judul "Density-based Clustering using Graphics Processors[1]", dengan kesimpulan bahwa DBSCAN dengan GPU membutuhkan waktu eksekusi 40 menit, jauh lebih cepat dibandingkan waktu eksekusi dengan CPU yang mencapai 3 jam. Pengujian ini dilakukan dengan menggunakan data dengan jumlah record sebanyak 1.000.000.

Tujuan dari penelitian ini untuk mengkombinasikan algoritma Cluster berbasis kepadatan dengan GPU, agar dapat memproses data dengan dimensi tinggi untuk menghasilkan akurasi tinggi dengan waktu penggalian data yang singkat.

2. Tinjauan Pustaka

a. Data Mining

Data mining merupakan cabang ilmu yang menggabungkan basis data, statistik, kecerdasan buatan dan machine learning. Contoh kasus dalam data mining seperti, pencarian nama yang paling sering dipakai di negara bagian Amerika Serikat atau mengelompokkan sejumlah dokumen dari hasil pencarian dengan search engine berdasarkan konteksnya[4]. Tujuan akhir dari data mining untuk mendapatkan informasi penting dari data mentah.

Tahap pertama penggalian data adalah input data, lalu dilanjutkan menuju tahap kedua yaitu preprocessing yang mana didalamnya terdapat proses feature selection, dimensionality reduction, dan normalization. Tujuan preprocessing adalah menyiapkan data masukan sebelum proses data mining. Kemudian pada tahap ketiga ada proses data mining yang didalamnya terdapat empat inti, yaitu predictive modeling, association analysis, Cluster analysis, dan anomaly detection. Pada tahap yang terakhir ada postprocessing yang merupakan hasil dari data mining.

b. Preprocessing

Preprocessing adalah tahapan yang bertujuan mempersiapkan data sebelum ditambang[10].

Sampling

Sampling adalah proses untuk melakukan seleksi terhadap data yang akan dianalisa. Sampling digunakan jika pengolahan pada data secara keseluruhan membutuhkan waktu dan biaya yang mahal. Data sample akan bekerja apabila datanya bersifat representatif [7].

Dimensionality Reduction

Dimensionality reduction merupakan proses eliminasi terhadap atribut yang tidak relevan ataupun menemukan atribut baru merujuk pada atribut data asli [6].

Discretization dan Binarization

Teknik ini bertujuan untuk mengubah atribut menjadi bentuk categorical [6].

Missing Value

Penanganan missing value adalah dengan mengganti nilai yang hilang dengan nilai yang sering muncul (modus) ataupun diganti dengan nilai rata-rata (mean).

Partisi Data

Partisi data merupakan proses untuk membagi data keseluruhan menjadi dua bagian, yaitu : data untuk pelatihan dan data untuk pengujian.

c. Clustering

Clustering merupakan pengelompokkan data dengan kedekatan karateristik kedalam suatu kelompok (Cluster) dengan aturan tertentu[3].Jika pada klasifikasi sifat/ karakteristik data ditentukan sebelumnya, pada Clustering sifat data tidak demikian, melainkan terlihat dari pengelompokkan data tersebut.

d. Similarity and Distance Measures

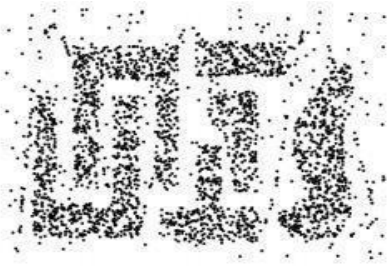
Similarity Measure merupakan jarak kedeketan suatu data dengan yang lainnya, sedangkan Distance Measure jarak perbedaan satu data dengan yang lain[9]. Pada Clustering terdapat data numerik dan data nominal. Untuk data numerik perhitungan jarak dapat dilakukan dengan formula Euclidean distance.

$$d(i,j) = \sqrt{(xi_1 - xj_1)^2 + (xi_2 - xj_2)^2 + \cdots + (xi_n - xj_n)^2}$$

Penggunaan euclidean distance pada Clustering sudah banyak ditemukan. Hal ini dikarenakan Euclidean distance mudah dipahami dan sensitif terhadap pencilan[6].

e. Algoritma Density-Based Clustering

Density-Based Clustering mengelompokkan data berdasarkan kepadatannya. Pada algoritma Density-Based, Cluster berisi kumpulan data dengan kepadatan yang tinggi dan terpisah dengan Cluster yang berisi data dengan kepadatan rendah[1,4,9]. Keuntungan dari algoritma ini, yaitu dapat mendeteksi Cluster dengan bentuk yang tidak tetap, sensitif terhadap pencilan, dan memilki akurasi yang lebih baik dibandingkan algoritma partitioning dan herarchy Clustering[7,9].



Gambar 2. 1 Salah satu hasil pengelompokkan dengan Density-Based Clustering[8]

f. Density Based Spatial Clustering of Application with Noise (DBSCAN)

Pseudo Code DBSCAN[1,7,9]

```
Input (Data, eps, minPoint) {
    Cluster = 0                  /* set cluster awal = 0
    for tiap titik P di dataset Data {
        if P dikunjungi lanjutkan ke titik selanjutnya
            tandai sudah dikunjungi
        hitung neighborPts          /*perhitungan jarak
menggunakan Euclidean distance
            if ukuran dari neighborPts < minPoint
               tandai P sebagai noise
            else {
               C = next cluster    /*membentuk cluster
selanjutnya
               expandCluster (P, neighborPts, C, eps,
minPoint)      /*memanggil fungsi expandCluster
               }
    }
}
expandCluster(P, neighborPts, C, eps, minPoint){
   tambahkan titik P ke cluster C
   for tiap titik P' di neighborPts {
      if P' sudah dikunjungi
         hitung neighborPts'
            if ukuran dari neighborPts >= minPoint
               neighborPts = neighborPts bergabung dengan
neighborPts'
      if P' belum tergabung ke cluster manapun
         tambahkan P' ke cluster C
   }
}
return semua cluster          /*menampilkan semua cluster
```

g. Graphics Processing Unit (GPU)

GPU biasanya dikembangkan untuk grafis sebuah game, namun GPU saat ini juga dikembangkan untuk tujuan komputasi dan aplikasi grafis lainnya. Dari segi hardware, GPU memiliki sejumlah multiprocessor, dan setiap processor memiliki SIMD-processor (Single Instruction Multiple Data)[1]. SIMD artinya dengan satu instruksi dapat mengeksekusi sejumlah data paralel dalam waktu yang bersamaan. GPU menghasilkan peformansi yang signifikan untuk proses eksekusi yang dijalankan[1]. NVIDIA, salah satu perusahaan yang meyediakan platform berbasis GPU mengenalkan teknologi Compute Unified Device Architecture (CUDA). Teknologi ini merupakan sebuah tools untuk bahasa pemograman C, digunakan untuk mengoperasikan program host dan program kernel secara bersamaan. Program host atau dikenal dengan program utama dieksekusi oleh CPU, sedangkan program kernel dieksekusi secara paralel oleh ratusan prosesor pada GPU[1]. Dengan CUDA inilah eksekusi program DBSCAN dapat berjalan pada GPU untuk menghasilkan biaya komputasi yang minimum.

Pseudo code untuk paralel DBSCAN dengan GPU[7].

```
Input (Data, eps, minPoint) {
     Cluster = 0          /* set cluster awal = 0

do in parallel{
     for tiap titik P di dataset Data {
          if P dikunjungi lanjutkan ke titik selanjutnya
               tandai sudah dikunjungi
     hitung neighborPts /*perhitungan jarak menggunakan
Euclidean distance dikerjakan di GPU
}end parallel

          if ukuran dari neighborPts < minPoint
                    tandai P sebagai noise
          else {
          C = next cluster /*membentuk cluster
selanjutnya
          expandCluster (P, neighborPts, C, eps,
minPoint)        /*menanggil fungsi expandCluster
                   }
     }

}

expandCluster(P, neighborPts, C, eps, minPoint){
   tambahkan titik P ke cluster C
   for tiap titik P' di neighborPts {
      if P' sudah dikunjungi
         hitung neighborPts'
               if ukuran dari neighborPts >= minPoint
                  neighborPts = neighborPts bergabung
dengan neighborPts'
      if P' belum tergabung ke cluster manapun
         tambahkan P' ke cluster C
   }
}

return semua cluster              /*menampilkan semua
cluster
```

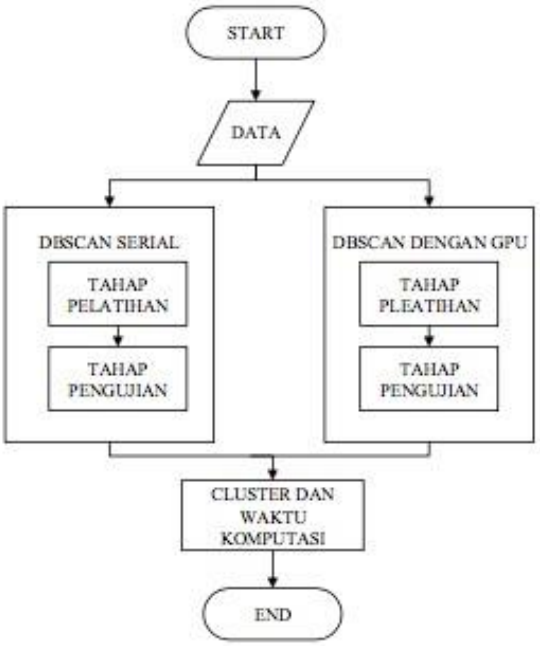
h. JCUBLAS

CUBLAS library merupakan implementasi dari Basic Linear Algebra Subprograms (BLAS). JCUBLAS memungkinkan programmer mengakses sumber daya pada GPU sehingga dapat melakukan dasar perhitungan matriks pada bahasa pemograman Java.

BLAS memiliki 3 level dengan fungsi tertentu. BLAS level-1 merupakan fungsi yang melakukan operasi berbasis skalar dan vektor. BLAS level-2 merupakan fungsi yang melakukan operasi matriks- vektor. BLAS level-3 merupakan fungsi yang melakukan operasi matriks-matriks.

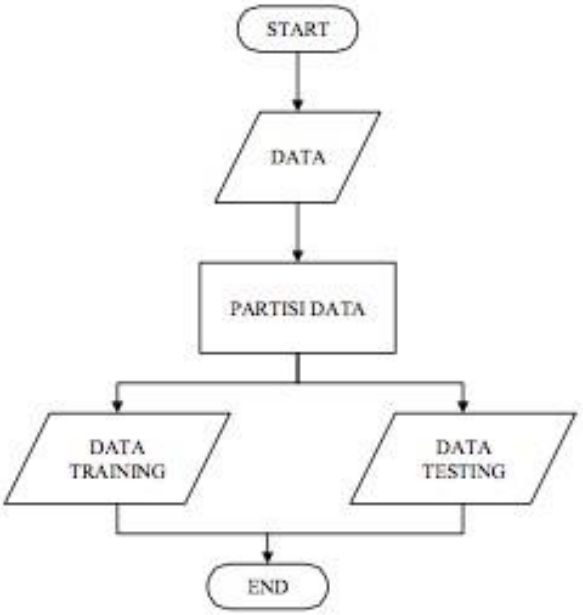
3. Perancangan Sistem

Gambaran umum



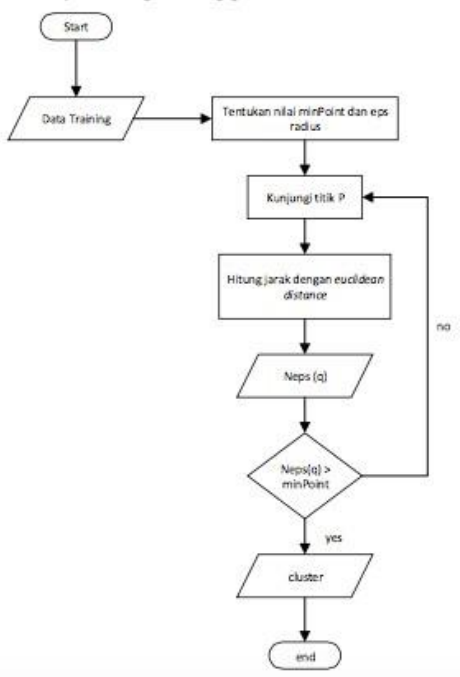
Gambar 3.1 Gambaran Umum

Preprocessing



Gambar 3.2 flowchart preprocessing

Flowchart pelatihan

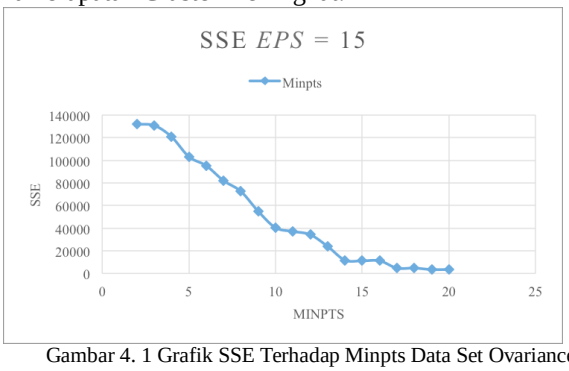


Gambar 3.3 flowchart pengujian DBSCAN

4. Analisis dan Pembahasan

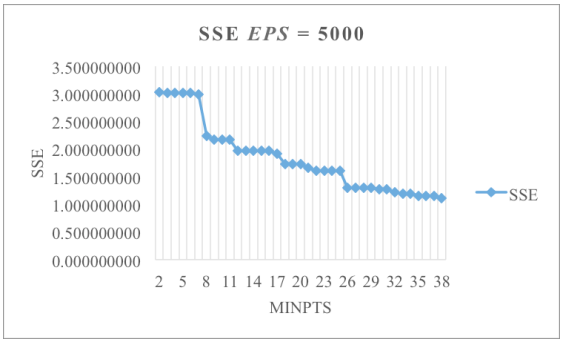
a. Analisis *Sum Square Error* (SSE)

Pada penelitian ini, untuk menentukan parameter Eps dan Minpts dilakukan dengan trial and error hingga didapatkan SSE terkecil. Semakin besarnya nilai Minpts menyebabkan perubahan yang signifikan terhadap nilai SSE. Hal ini dikarenakan Minpts menjadi syarat minimum terbentuknya Cluster. Semakin besar nilai Minpts menyebabkan nilai SSE yang semakin kecil karena kerapatan Cluster meningkat.



Gambar 4. 1 Grafik SSE Terhadap Minpts Data Set Ovariance

Pada grafik diatas terlihat dengan nilai epsilon 15, setelah dilakukan beberapa pergantian nilai Minpts, SSE terbaik pada didapatkan dengan Minpts terbesar 20.



Gambar 4. 2 Grafik SSE Terhadap Minpts Data Set Prostate

b. Analisis Waktu

Pada data set *prostate cancer*, waktu eksekusi untuk serial dan paralel berada di bawah 1 detik. Hal ini dikarenakan dimensi data set *prostate cancer* lebih sedikit dibanding dengan data set *ovariance cancer*. Selain itu berdasarkan uji SSE, data set *prostate cancer* merupakan contoh data set yang baik untuk algoritma DBSCAN, dikarenakan nilai SSE yang didapatkan rata – rata dibawah 5. Faktor lainnya yang menyebabkan waktu eksekusi data set *prostate cancer* lebih cepat dikarenakan DBSCAN mampu menentukan *seed* yang tepat tanpa harus menelusuri keseluruhan isi data.

Waktu eksekusi pada GPU juga menghasilkan performa yang lebih baik dibandingkan dengan serialnya untuk kedua data set. Pada data set *ovariance cancer* terdapat perbedaan waktu eksekusi sebesar 1.035921875 detik dan 0.063893878 detik untuk data set *prostate cancer*.

c. Analisis Performansi

GPU memungkinkan percepatan terhadap pencarian jarak antar *record* pada data. Hal ini mengurangi biaya komputasi dalam hal waktu eksekusi. Namun disisi lain penyediaan kartu grafis yang lebih baik membutuhkan biaya yang lebih banyak pula. Untuk itu pengukuran performansi digunakan untuk mengetahui efisiensi dari penggunaan gpu untuk algoritma DBSCAN.

5. Kesimpulan dan Saran

a. Kesimpulan

Kesimpulan dari tugas akhir ini, yaitu :

1. Cara mengimplementasikan algoritma *Density Based Spatial Clustering Application with Noise (DBSCAN)* menggunakan GPU adalah dengan memparalelkan perhitungan *Euclidean distance*[13]. Perhitungan ini dapat dilakukan dengan menggunakan library JCUBLAS pada Java untuk proses perkalian matriks yang terdapat dalam persaman *Euclidean distance.*
2. Untuk data set *Ovarian Cancer* didapatkan hasil SSE terkecil 179.0626057 dengan *epsilon* 10 dan *Minpts* 3. Sedangkan untuk data set *Prostate cancer* didapatkan SSE terkecil 1.111533174 dengan *epsilon* 5000 dan *minpts* 38. Sedangkan untuk waktu komputasi pada GPU dan serial memiliki perbedaan 1.035921875 s untuk data set *Ovarian Cancer* serta 0.063893878 s untuk data set *Prostate cancer*.
3. Pada analisa peformansi untuk *Speedup* dan *Performance Improvement*, pada kedua data set memiliki nilai tertinggi dengan jumlah *thread* enam. Untuk efisiensi didapatkan nilai terbaik dengan jumlah *thread* tiga.

b. Saran

1. Implementasi DBSCAN pada GPU dapat menggunakan bahasa pemograman lain seperti phyton atau C++ yang berbasis Object Oriented Programming dan dapat diimpelementasikan dengan CUDA.
2. Untuk penelitian lebih lanjut dapat menggunakan perangkat keras yang lebih baik, seperti super computer atau NVIDIA GEFORCE seri 900M ke atas.

Daftar Pustaka:

[1] Böhm, Christian, Noll, Robert, Plant, Claudia, Wackersreuther, B 2009, Density-based Clustering using Graphics Processors, University of Munich, Munich, Germany.

[2] Budiarti, A 2006, 'Aplikasi dan Analisis Clustering pada Data Akademik', Universitas Indonesia, Jakarta, Indonesia.

[3] Cao, F, Ester, M, Qian, W, Zhou, A, Density-Based Clustering over an Evolving Data Stream with Noise, Department of Computer Science and Engineering, Fudan University, Shanghai, China.

[4] Dunham, H M(2003), Data Mining Introductory and Advanced Topic, New Jersey: Pearson Education Inc.

[5] Ester, M, Kriegel, Hans-Peter, Sander, J, Xu, X 2005, A Density-Based Algorithm for Discovering Cluster in Large Spatial Databases with Noise, Institute for Computer Science, University of Munich Oettingenstr, 67, D- 80538 München, Germany.

[6] Jiang, D, Jian, Pei, Zhang, A, DHC: A Density-based Hierarchical Method for Time Series Gene Expression Data, Department of Computer Science and Engineering, State University of New York, Buffalo, USA.

[7] Nagpai, Pooja , Mann, Priyanka A 2011, Comparative Study of Density based Clustering Algorithms, Department Of Computer Science, NIT Kurukshetra.

[8] Patwary, Md. M A, Palsetia, D, Agrawal, A, Liao, Wei-keng, Manne, F, Choudhary, A 2012,A New Scalable Parallel DBSCAN Algorithm Using the Disjoint-Set Data Structure, Northwestern University, Evanston, IL 60208, USA, University of Bergen, Norway.

[9] Rahmat, Kemas, Putro, Bima A, Shaufiah, 'Implementasi Density Based Spatial Clustering Application with Noise (DBSCAN) Dalam Perkiraan Terjadinya Banjir di Bandung', Fakultas Informatika, Institut Teknologi Telkom, Bandung.

[10] Sander, J, Ester, M, Kriegel, Hans-Peter, Xu, X, Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications, Institute for Computer Science, University of Munich Oettingenstr, 67, D- 80538 München, Germany.

[11] Welton, B, Samanas, E, Miller, B P 2013, Mr. Scan: Extreme Scalable Density-Based Clustering using a Tree-Based Network of GPGPU Nodes, Computer Science Department, University of Wisconsin Madison, WI 53706, Denver, Colorado, USA.

[12] Wang, Bingchen, Zhang, Chenglong, Song, Lei, Zhao, Lianhe, Dou, Yu, Yu, Zihao, Design and optimization of DBSCAN Algorithm based on CUDA, Chinese Academy of Sciences, Beijing, China.

[13] NVIDIA. (2015). "CUDA TOOLKIT DOCUMENTATION", [online], diambil dari situs (http://docs.nvidia.com/cuda/cublas/#axzz3ikL5kLWs, diakses tanggal 6 September 2015)

[14] NVIDIA. (2015). "CUDA TOOLKIT DOCUMENTATION", [online], diambil dari situs (http://docs.nvidia.com/cuda/cublas/#axzz3ikL5kLWs, diakses tanggal 6 September 2015)