

MODIFIKASI FUNGSI DENSITY PADA ALGORITMA ANT CLUSTERING

Kurniawan Nur Ramadhani¹⁾, Febryanti Sthevanie²⁾

Fakultas Informatika Universitas Telkom

Jln Telekomunikasi No. 1 Terusan Buah Batu Bandung 40257

¹⁾kurniawanr@telkomuniversity.ac.id, ²⁾sthevanie@telkomuniversity.ac.id

Abstrak

Clustering merupakan salah satu tugas dalam data mining untuk mengelompokkan data berdasarkan kemiripan karakteristik. Pada penelitian ini, akan diusulkan sebuah modifikasi pada algoritma Ant Clustering untuk mempercepat proses komputasi. Modifikasi dilakukan pada fungsi density dengan mempertimbangkan batasan pemisahan spasial. Dari hasil percobaan yang dilakukan dengan data sejumlah 800 baris dan jumlah iterasi sebanyak 1000, didapatkan bahwa modifikasi fungsi density pada algoritma Ant Clustering berhasil meningkatkan kecepatan dengan nilai akurasi yang tidak terlalu berbeda dengan algoritma Ant Clustering standar.

Kata kunci:

Clustering, Ant Clustering, fungsi density

Abstract

Clustering is one of the tasks in data mining to group data based on similar characteristics. In this study, will be proposed a modification on Ant Clustering algorithm to speed up the process of computing. Modifications carried on by considering the density function limits the spatial separation. From the results of experiments conducted with a number of data lines 800 and the number of iterations of 1000, it was found that the density modification function on Ant Clustering algorithms managed to increase the speed with accuracy values that are not too different from Ant Clustering algorithm standard.

Keywords:

Clustering, Ant Clustering, density function

I. PENDAHULUAN

Data mining adalah sekumpulan metode yang digunakan untuk mendapatkan informasi dari serangkaian data yang tersedia dengan jumlah yang banyak. Dalam *data mining* dikenal serangkaian metode untuk dapat mengekstrak informasi dari data berjumlah banyak atau data berdimensi tinggi dan bernilai heterogen. Terdapat dua kategori dalam *data mining*. Kategori pertama adalah kategori prediksi. Algoritma pada kategori ini digunakan untuk menentukan nilai sebuah variabel dari serangkaian data sebelumnya yang diketahui nilainya. Contoh algoritma ini adalah algoritma klasifikasi dan regresi. Kategori kedua adalah kategori deskriptif. Algoritma pada kategori ini digunakan untuk mendapatkan deskripsi dari serangkaian data yang dapat dimengerti oleh manusia. Algoritma dalam kategori ini adalah algoritma *clustering* dan asosiasi (J. Han and M. Kamber, 2001).

Clustering merupakan proses untuk mengelompokkan serangkaian data ke dalam satu kelompok berdasarkan kesamaan karakteristik dari data tersebut. Perbedaan *clustering* dengan klasifikasi adalah pada *clustering*, proses pengelompokkan dilakukan murni berdasarkan kesamaan karakteristik nilai atribut-atribut dalam data tersebut, sedangkan pada klasifikasi, data dikelompokkan berdasarkan kesamaan nilai dalam satu atribut. Algoritma *clustering* yang baik harus dapat memenuhi kriteria berikut (Jain Anil K, 1998):

1. *Scalability* yaitu mampu menangani data dalam ukuran yang kecil maupun besar.
2. Mampu menangani berbagai jenis tipe atribut.

3. Menemukan bentuk-bentuk *cluster* yang unik.
4. Parameter input tidak terlalu rumit sehingga tidak menyulitkan pengguna untuk mengontrol kualitas *cluster* yang terbentuk.
5. Mampu menangani *noise*.
6. Tidak terpengaruh urutan penyisipan *record*.
7. Mampu menangani data yang berdimensi tinggi.
8. Mudah dipahami dan mudah digunakan.

Dalam *clustering* terdapat beberapa model *cluster*. Model pertama adalah *centroid* model. Model *centroid* bekerja dengan mendefinisikan sebuah *cluster* menggunakan radius dan posisi *centroid* dalam ruang N-dimensi. Contoh algoritma ini adalah K-Means. Model kedua adalah *hierarchical* model. Model ini bekerja dengan menghasilkan *cluster* bersarang yang direpresentasikan sebagai pohon hirarki. Contoh algoritma ini adalah *agglomerative* dan *divisive clustering*. Model ketiga adalah *density* model. Model ini mendefinisikan *cluster* melalui sebuah nilai kepadatan *cluster*. Serangkaian data dikatakan berada pada satu *cluster* jika nilai kepadatannya memenuhi nilai tertentu. Contoh algoritma ini adalah DBSCAN dan Ant clustering (Jain Anil K, 1998).

Pada penelitian sebelumnya (Marco Dorigo and Thomas Stutzle, 2004) telah dilakukan proses analisis pada algoritma *Ant Clustering* yang diusulkan juga dalam penelitian (Monmarche N, 1999). Paper ini akan membahas proses modifikasi pada algoritma *Ant Clustering* dan hasil analisis terhadap modifikasi algoritma tersebut. Algoritma dimodifikasi pada bagian *neighbourhood function* dengan harapan proses *clustering* menjadi lebih cepat dengan nilai akurasi yang tidak terlalu jauh berbeda dengan algoritma awal.

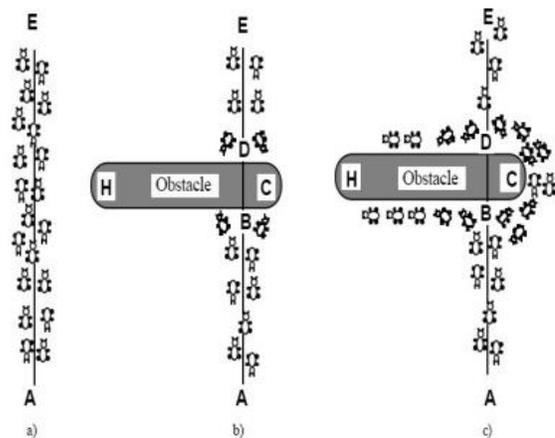
II. KAJIAN LITERATUR

II.1 Algoritma *Ant Colony*

Ant Colony Optimization termasuk teknik pencarian multi agent yang sering digunakan untuk permasalahan optimasi, khususnya kombinatorial, yang terinspirasi oleh tingkah laku semut dalam suatu koloni. Algoritma ACO pertama kali diperkenalkan oleh Marco Dorigo pada tahun 1991 yang kemudian dipublikasikan dengan nama *Ant System (AS)* (Marco Dorigo and Thomas Stutzle, 2004). Perilaku semut yang cukup menarik adalah ketika mereka mencari makan. Semut dapat menemukan jalur terpendek

antara sumber makanan dan sarang mereka. Ketika berjalan dari sumber makanan ke sarang dan sebaliknya, semut meletakkan suatu zat (yang disebut *feromon*) di sepanjang jalur yang mereka lalui. Ketika zat tersebut disekresikan sebagai isyarat seekor semut, maka semut yang lain dapat mengenalinya.

Ketika mencari makan, pada awalnya semut akan berkeliling di daerah sekitar sarangnya secara acak. Begitu mengetahui ada makanan, semut akan menganalisa kualitas dan kuantitas makanan tersebut dan membawa beberapa bagian ke sarangnya. Dalam perjalanannya, mereka meninggalkan jejak berupa sejumlah zat kimia, yang disebut *feromon*. *Feromon* ini akan membimbing semut lain untuk menemukan sumber makanan. Jumlah *feromon* yang ditinggalkan oleh semut bergantung pada jumlah makanan yang ditemukan. Semakin banyak makanan yang didapat, semakin banyak pula jumlah *feromon* yang ditinggalkan. Sehingga semakin banyak semut yang melewati suatu jalur, semakin kuat pula jejak *feromon* yang terkumpul di jalur tersebut.



Gambar 1. Ilustrasi *Ant Colony*

Gambar 1 menunjukkan perjalanan semut berjalan dari titik A ke titik E. Pada awalnya ketika belum diberikan pembatas, maka semut akan berjalan dengan jumlah yang sama di sebelah kiri dan kanan garis pembatas. Ketika diberikan penghalang, maka semut pada awalnya akan sama. Akan tetapi, lama kelamaan semut akan cenderung melewati sebelah kanan garis pembatas karena jarak yang ditempuh

lebih pendek. Hal itu dikarenakan pengaruh *feromon* tadi (Marco Dorigo and Thomas Stutzle, 2004).

II.2 Ant Clustering

Penelitian yang dilakukan pada (Monmarche N, 1999) memodifikasi algoritma *Ant Colony* agar dapat melakukan operasi *clustering* pada serangkaian data yang diberikan. Salah satu modifikasi yang mencolok adalah tidak adanya konsep feromon. Namun, yang menjadi pemicu dari kerja algoritma semut adalah tingkat kepadatan (*density*) dari item yang tersebar pada grid. Algoritma ini berjalan dalam sebuah peta topografi. Dalam peta tersebut, data dan agen semut disebar secara acak. Di awal, setiap agen semut mengambil data secara acak yang tersebar pada peta. Kemudian semut akan bergerak secara acak pada arah tertentu sejauh langkah yang sudah didefinisikan sebelumnya. Kemudian, semut akan menghitung probabilitas dari data tersebut akan diletakkan pada koordinat semut atau tidak. Nilai probabilitas ini disebut p_{drop} . Jika data tersebut diletakkan, maka agen semut akan mencari data baru lagi untuk diambil. Probabilitas dari seekor agen semut untuk mengambil data baru disebut p_{pick} . Proses ini akan dilakukan secara berulang hingga nilai iterasi tertentu. Secara umum, algoritma semut untuk proses *Clustering* dapat dituliskan sebagai berikut (Monmarche N, 1999):

```
begin
INITIALISATION PHASE
Randomly scatter data items on the
toroidal grid
for each j in 1 to #agents do
  i:=random_select(rem_items)
  pick_up(agent(j),i)
  g:=random_select(rem_empty_loc)
  place_agent(agent(j),g)
end for
MAIN LOOP
for each it_ctr in 1 to #iterations
do
  j:=random_select(all_agents)
  step(agent(j),stepsize)
  i:=carried_item(agent(j))
  drop:=drop_item?(f(i))
  if drop=TRUE then
    while pick=FALSE do
      i:=random_select(free_
items)
      pick:=pick_item?(f(i))
    end while
  end if
end for
```

end

Nilai dari p_{drop} dan p_{pick} dirumuskan sebagai berikut.

$$p_{drop}(i) = \left(\frac{f(i)}{k_d + f(i)} \right)^2 \quad [1]$$

$$p_{pick}(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2 \quad [2]$$

dengan

k_d = koefisien *drop probability*

k_p = koefisien *pick probability*

Kedua nilai tersebut memiliki rentang nilai 0 hingga 1. Nilai $f(i)$ merupakan nilai fungsi kepadatan yang dihitung menggunakan persamaan berikut.

$$f(i) = \max \left(0, \frac{1}{|S|} \sum_{\{c_j \in S | d_{ij} \neq nil\}} \left(1 - \frac{d(k, d_{ij})}{\alpha \mu} \right) \right) \quad [3]$$

dengan

d_{ij} = indeks dari dokumen pada tabel.

$|S|$ = ukuran ketetanggaan di sekeliling dokumen.

$d(k, d_{ij})$ = nilai *dissimilarity* antara dokumen k yang dibawa semut dengan dokumen d_{ij} .

N = jumlah total dokumen.

μ = faktor pengali

Nilai μ dihitung dari persamaan berikut.

$$\mu = \frac{2}{N(N-1)} \sum_{k=1}^N \sum_{l=1}^{k-1} d(k, l) \quad [4]$$

dan nilai α antara 0 hingga 1.

II.3 Modifikasi Fungsi Density

Penelitian yang dilakukan pada (Handl, Julia, Knowles and Dorigo, 2006) mengusulkan modifikasi pada fungsi kepadatan (*density*). Modifikasi ini dirumuskan pada persamaan berikut.

$$f(i) = \left\{ \begin{array}{l} \frac{1}{|S|} \sum_{\{c_j \in S | d_{ij} \neq nil\}} \left(1 - \frac{d(k, d_{ij})}{\alpha \mu} \right), \forall j \left(1 - \frac{d(k, d_{ij})}{\alpha \mu} > 0 \right) \\ 0, \exists j \left(1 - \frac{d(k, d_{ij})}{\alpha \mu} \leq 0 \right) \end{array} \right\} \quad [5]$$

Tujuan modifikasi ini adalah meningkatkan kecepatan eksekusi dari algoritma *Ant Clustering*.

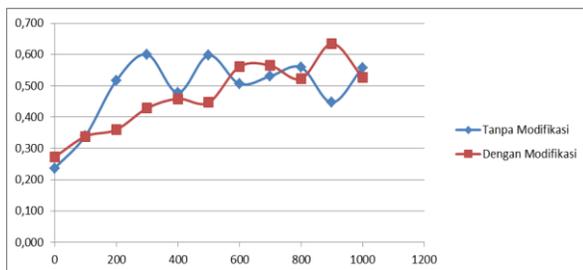
Batasan $\forall j \left(1 - \frac{d(k, d_{ij})}{\alpha \mu} > 0 \right)$ berfungsi sebagai

pemisah untuk nilai data dengan nilai *dissimilarity* yang tinggi, yang akan meningkatkan pemisahan spasial antar *cluster*.

III. ANALISIS

Pengujian dilakukan dengan membandingkan kinerja antara algoritma *Ant Clustering* standar dengan algoritma yang telah mengalami modifikasi pada fungsi *density*. Kinerja yang diukur ada dua, yakni nilai korelasi *Pearson* dan waktu eksekusi. Nilai korelasi *Pearson* menggambarkan keterkaitan antara pemetaan data yang dilakukan agen semut pada peta topografi dengan karakteristik data. Nilai korelasi *Pearson* berkisar dari -1 hingga 1. Semakin mendekati 1, semakin tinggi keterkaitan antara pemetaan data pada peta topografi dengan karakteristik data. Artinya, nilai korelasi *Pearson* yang mendekati 1 jika data pada peta topografi berhasil dikelompokkan pada satu kelompok dengan nilai *density* minimal.

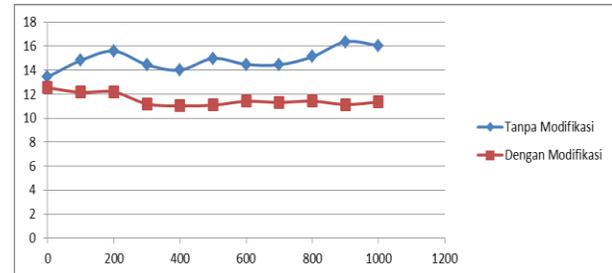
Pengujian dilakukan dengan menggunakan data set sebanyak 800 *record*. Algoritma dijalankan selama 1000 iterasi. Nilai korelasi *Pearson* dari kedua algoritma dapat dilihat pada grafik seperti pada gambar 2.



Gambar 2. Perbandingan Nilai Korelasi Pearson

Dari grafik tersebut, dapat dilihat bahwa nilai korelasi *Pearson* dari kedua algoritma tidak terlalu berbeda. Secara umum, nilai korelasi *Pearson* dari algoritma awal (tanpa modifikasi) pada awal iterasi lebih tinggi. Namun mendekati akhir iterasi, nilai

korelasi *Pearson* tidak terpaut jauh. Adapun perbandingan waktu eksekusi kedua algoritma dapat dilihat pada grafik seperti pada gambar 3.



Gambar 3. Perbandingan Waktu Eksekusi

Dari grafik di atas, dapat dilihat bahwa waktu eksekusi algoritma *Ant Clustering* yang dimodifikasi selalu lebih rendah dibandingkan algoritma *Ant Clustering* standar. Hal ini menunjukkan bahwa modifikasi fungsi *density* berhasil meningkatkan kecepatan dari algoritma *Ant Clustering*.

IV. KESIMPULAN

Dalam penelitian ini dilakukan modifikasi fungsi *density* pada algoritma *Ant Clustering*. Proses modifikasi ini dilakukan dengan meningkatkan pemisahan spasial pada nilai *dissimilarity* yang tinggi. Modifikasi ini juga mempercepat proses perhitungan *density* sehingga mempercepat algoritma *Ant Clustering* secara keseluruhan. Dari hasil eksperimen yang dilakukan, modifikasi fungsi *density* ini berhasil meningkatkan kecepatan algoritma *Ant Clustering* dengan nilai akurasi menggunakan korelasi *Pearson* yang tidak jauh beda dengan nilai akurasi menggunakan algoritma *Ant Clustering* standar.

REFERENSI

- J. Han dan M. Kamber, 2001, "Data Mining: Concepts and Techniques", CA: Morgan Kaufmann, San Francisco.
- Jain Anil K, 1998, "Algorithms for Clustering Data", Prentice Hall, New Jersey.1998.

Marco Dorigo and Thomas Stutzle, 2004, "Ant Colony Optimization. The MIT Press, Massachusetts.

Monmarche N, 1999, "On data clustering with artificial ants. In: Freitas, A.A. (Ed.) Data Mining with Evolutionary Algorithms: Research Directions—Papers from the AAAI Workshop. AAAI Press, pp 23-26. 1999.

Handl, Julia, Knowles dan Dorigo, 2006,"Ant Based Clustering and Topographic Mapping. In: Artificial Life volume 12. MIT, pp 35-61.