DATA MINING UNTUK KLASTERING DATASET MULTIDIMENSIONAL MENGGUNAKAN ALGORITMA ANT COLONY OPTIMIZATION

David
Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jln. H. Agus Salim N0. 72-76 Pontianak, Kalimantan Barat
Pos-el: David_Liauw@yahoo.com dan DavidLiauw@gmail.com

Abstracts: *Right now data mining activities are still hidden to further processed into useful knowledge in decision making. The process of information extraction from the collection of data that is stored with the data mining. One of the methods in Data Mining is clustering. Clustering aims to segment the physical or abstract objects in the form of classes or objects similar to the One is ACO (Ant Colony Optimization). On this research analyzed the influence of the changes made on the application of the level of accuracy and the resulting clusters, and system parameters that apply. For the purpose of software is built as a media test ACO algorithm. Level of accuracy can be improved by using pheromone and increase the value of the parameter and the maximum number of ant Iterations. But the more and the maximum number of ant addition, with improved Iterations then processing time will be longer.*

Keywords: *Data Mining, Clustering, Ant Colony Optimization, Cluster Analysis.*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini kian hari kian pesat, sehingga mendorong tingginya kebutuhan informasi. Informasi dapat diperoleh dari suatu timbunan data-data yang berukuran sangat besar. Data tersebut menjadi sumber data yang terbesar dan sangat berharga untuk setiap pengguna karena didalamnya terdapat kumpulan informasi yang saling terhubung. Setiap kali informasi ditambahkan, makin besar ukuran data dan semakin sulit pula untuk mencari informasi yang benar-benar yang diinginkan oleh karena itu dibutuhkan suatu teknologi untuk mendapatkan informasi yang benar-benar diinginkan tersebut.

Informasi yang didapat dari data sangat sedikit yang dapat dimanfaatkan, oleh karena itu perlu adanya aktivitas penggalian (ekstraksi) data yang masih

tersembunyi untuk selanjutnya diolah menjadi pengetahuan yang bermanfaat dalam pengambilan keputusan. Proses ekstraksi informasi dari kumpulan data-data yang tersimpan disebut dengan *data mining.*

Turban dan Aronson (1998) mengemukakan bahwa *data mining* merupakan suatu bentuk yang biasa digunakan untuk menggambarkan *knowledge discovery* dari sebuah database, *knowledge extraction, data archaeology, data exploration, data pattern processing, information harvesting* dan *software*. Jadi Data mining adalah suatu proses analisis data dari berbagai perspektif yang berbeda dan memberikan ringkasannya dalam bentuk informasi yang bermanfaat, memungkinkan pemakai untuk melakukan analisis data dari dimensi yang berbeda-beda, dan secara teknis merupakan proses menemukan korelasi atau bentuk dari suatu kumpulan database relasional. Salah satu penerapan teknik *data mining* adalah *clustering.* Banyak sekali metode *clustering* yang dapat diimplementasikan, namun penulis menggunakan algoritma *Ant colony optimization* untuk klastering.

Dari latar belakang yang dikemukakan diatas, maka yang menjadi rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan sebuah aplikasi *clustering* dokumen menggunakan algoritma *Ant colony optimization* menggunakan bahasa *Java*

Batasan masalah yang digunakan dalam penelitian ini adalah: 1) Penelitian dikonsentrasikan pada proses *clustering,* dan 2) Implementasi dari sistem *clustering* menggunakan algoritma *Ant colony optimization.*

Tujuan penelitian ini adalah membuat aplikasi clustering dokumen untuk perpustakan digital dengan mengunakan algoritma *Ant colony optimization* sebagai bagian dari teknologi *data mining* menggunakan bahasa Java.

2. TINJAUAN PUSTAKA

2.1 *Data Mining*

Data mining merupakan teknologi yang menggabungkan metoda analisis tradisional dengan algoritma yang canggih untuk memproses data dengan volume besar. Data mining adalah suatu proses mengolah data-data yang ada dengan metode dan algoritma tertentu, yang bertujuan mengekstrak pola yang berguna dari data yang besar tersebut (Han dan Micheline, 2001).

Adapun tugas data mining antara lain (Han dan Micheline, 2001): 1) Metoda prediksi, yaitu menggunakan beberapa variable untuk memperkirakan suatu nilai yang tidak diketahui dari variable yang lain. Sasaran pada tugas ini adalah memprediksikan nilai atribut tertentu berdasarkan nilai atribut yang lain. Atribut

yang diprediksi dikenal sebagai target atau variabel yang tergantung pada variabel lain, atribut yang digunakan selama membuat prediksi dikenal sebagai penjelasan (*explanatory*) atau variabel yang bebas. Contohnya antara lain *Classification, Regression* dan *Deviation* Detection, dan 2) Metoda deskripsi, yaitu mencari suatu pola yang dapat ditafsirkan manusia sehingga data dapat digambarkan atau diuraikan. Sasaran pada tugas ini adalah memperoleh pola (kecenderungan korelasi, *cluster* dan anomali) yang menyimpulkan hubungan dalam data. Tugas deskriptif *data mining* memerlukan teknik *postprocessing* untuk validasi dan kejelasan hasil. Contohnya antara lain *Clustering* dan *Association Rule Discovery.*

2.2 *Clustering*

Pada dasamya *clustering* terhadap data adalah suatu proses untuk mengelompokkan sekumpulan data tanpa suatu atribut kelas yang telah didefinisikan sebelumnya, berdasarkan pada prinsip konseptual *clustering* yaitu memaksimalkan dan juga meminimalkan kemiripan intra kelas (Ming dan Hou, 2004). Misalnya, sekumpulan obyek-obyek komoditi pertama-tama dapat di clustering menjadi sebuah himpunan kelas-kelas dan lalu menjadi sebuah himpunan aturan-aturan yang dapat diturunkan berdasarkan suatu klasifikasi tertentu.

Proses untuk mengelompokkan secara fisik atau abstrak obyek-obyek ke dalam bentuk kelas-kelas atau obyek-obyek yang serupa, disebut dengan *clustering* atau *unsupervised classification*. Melakukan analisa dengan *clustering*, akan sangat membantu untuk membentuk partisi-partisi yang berguna terhadap sejumlah besar himpunan obyek dengan didasarkan pada prinsip *divide and conquer* yang mendekomposisikan suatu sistem skala besar, menjadi komponen-komponen yang lebih kecil, untuk menyederhanakan proses desain dan implementasi.

Pada dasarnya terdapat dua tipe clustering, yaitu: 1) *Partitional Clustering*: Tipe cluster yang benar-benar terpisah antara sekelompok obyek dengan sekelompok obyek lainnya dan 2) *Hierarchical clustering*: Sekelompok clustr yang terorganisasi sebagai suatu pohon hirarki (*hierarchical tree*).

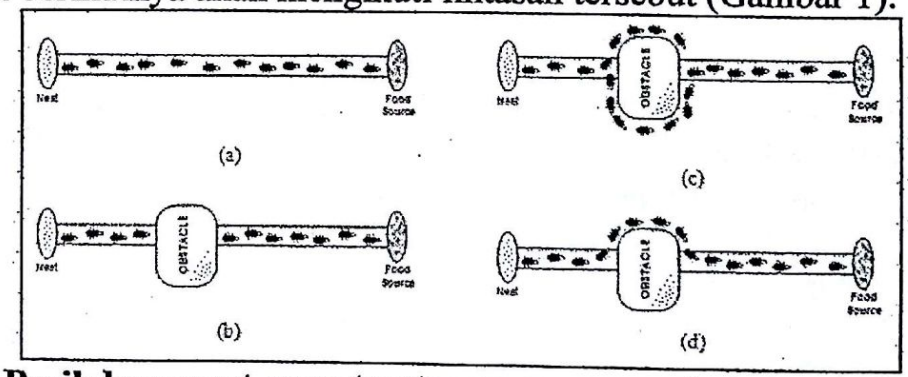
2.3 *Swarm Intelligence*

Menurut Dorigo, dkk (1999) dan Kennedy, dkk (2001) *Swarm Intelligence* adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (*collective behaviour*) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (*self-organizing*). Sistem yang memanfaatkan Swarm

Intelligence biasanya merupakan sebuah populasi yang terdiri atas anggota berupa agen-agen yang sederhana, yang berinteraksi secara lokal dengan sesama anggota, dan juga berinteraksi dengan lingkungan. Walaupun pada umumnnya tidak ada struktur kendali secara terpusat (*centralized*) yang mendikte bagaimana masing-masing individu bertindak, namun interaksi secara lokal (di antara anggota) seringkali menuju pada pembentukan (*emergence*) perilaku global. Contoh dari sistem seperti ini dapat ditemukan pada alam, misalnya koloni semut, burung-burung, kawanan rusa, bakteri, maupun ikan.

2.3.1 Koloni Semut (*Ant Colony*)

Ant Colony adalah salah satu penerapan *Swarm Intelligence* (Dorigo, dkk, 1999). Pada kehidupan sebenarnya, semut-semut meninggalkan sarang untuk mencari makanan dan harus mencari kembali sarang mereka. Misalkan ada segerombolan semut yang mencari makanan, maka semut yang berada di depan harus memilih lintasan tertentu untuk dilewati. Pada saat semut pertama bejalan, semut tersebut meninggalkan hormon feromon yang dapat dicium oleh semut berikutnya, sehingga semut-semut berikutnyatahu apakah tempat tersebut sudah dilewati atau belum. Semut yang melewati lintasan yang lebih pendek akan meninggalkan aroma feromon yang lebih tajam daripada semut yang menempuh lintasan yang lebih panjang. Hal ini terjadi karena feromon yang ditinggalkan dapat menguap. Saat semut-semut yang di belakang mengikuti semut-semut yang di depannya, semut-semut tersebut akan memilih lintasan berdasarkan kuatnya aroma feromon dan jarak lintasan. Semakin banyak semut yang menempuh suatu lintasan tertentum, maka aroma feromon pada lintasan tersebut akan semakin kuat sehingga semut-semut berikutnya akan mengikuti lintasan tersebut (Gambar 1).



Gambar 1. Perilaku semut-semut antara sarangnya dengan sumber makanan, (a) Semut-semut mengikuti jalur antara sarangnya dengan sumber makanan, (b) Semut-semut menghadapi suatu penghalang, (c) Pemilihan jalur-jalur yang dilakukan oleh semut-semut, (d) Semut-semut menemukan jalur terpendek

2.3.2 *Ant Colony Optimization*

Algoritma *Ant Colony Optimization* (*ACO*) ialah sebuah sistem yang berdasarkan agen-agen yang mensimulasikan perilaku dari semut-semut, termasuk mekanisme kerja sama dan adaptasi (Dorigo, dkk, 1999).

Algoritma ACO dibuat berdasarkan gagasan berikut: 1) Setiap jalur yang dilalui oleh semut diasosiasikan dengan kandidat solusi dari suatu masalah, 2) Jika suatu semut mengikuti suatu jalur, jumlah pheromone pada jalur tersebut sebanding dengan kualitas kandidat solusi yang bersangkutan, 3) Jika suatu semut diharuskan untuk memilih antara dua jalur, jalur yang memiliki jumlah pheromone lebih banyak memiliki peluang lebih besar untuk dipilih semut tersebut.

Hasilnya, semut-semut akan memusat ke jalur yang pendek, dengan harapan menghasilkan solusi yang optimal atau mendekati optimal. Intinya, desain algoritma ACO berdasarkan hal-hal berikut (Dorigo dkk, 1999): 1) Dapat merepresentasikan masalah dengan baik, dimana semut-semut secara bertahap membangun dan mengubah solusi dengan menggunakan aturan transisi probabilistik, berdasarkan jumlah pheromone pada suatu jalur dan sebuah fungsi heuristik, 2) Sebuah fungsi heuristik yang sesuai dengan masalah (η) yang menyatakan kualitas item yang akan ditambahkan ke solusi sementara, 3) Aturan dalam melakukan update jumlah pheromone, yang menjelaskan bagaimana mengubah jumlah pheromone (τ) suatu jalur, dan 4) Aturan transisi probabilistik berdasarkan nilai fungsi heuristik (η) dan jumlah pheromone (τ) yang akan digunakan dalam membangun suatu solusi.

2.3.3 *Clustering* Menggunakan *Ant Colony Optimization*

Dalam algoritma ACO setiap semut secara bertahap membangun/mengubah sebuah solusi dari masalah yang diberikan, pada kasus ini adalah permasalahan clustering.

Permasalahan clustering data dimodelkan sebagai suatu masalah optimasi clustering. Diberikan suatu himpunan data yang terdiri dari m obyek data dengan n atribut dan ditentukan sejumlah cluster (g). Persamaan (1) menyatakan fungsi obyektif. Persamaaan (3) menyatakan bahwa setiap obyek data hanya untuk satu *cluster* (Kao dan Cheng, 2006).

$$J(W,C) = \sum_{i=1}^{m}\sum_{j=1}^{g} w_{ij} \left\| X_i - C_j \right\| \quad \ldots (1)$$

Dimana $\|X_i - C_j\| = \sqrt{\sum_{v=1}^{n}(x_{iv} - c_{jv})^2}$ dan $\sum_{j=1}^{g} w_{ij} = 1, \quad i = 1,...,m$

Jika data *i* termasuk ke dalam cluster *j* maka $w_{ij} = 1$, jika tidak $w_{ij} = 0$.

$$C_j = \frac{\sum_{i=1}^{m} w_{ij} X_i}{\sum_{i=1}^{m} w_{ij}}, j = 1,...,g \quad \ldots (2)$$

Keterangan:

x_i : Vektor data obyek ke-*i* dan $x_i \in R^n$

x_{iv} : Nilai atribut ke-*v* dari obyek data ke-*i*

c_j : Vektor dari pusat cluster ke-*j* dan $c_j \in R^n$

c_{jv} : Nilai dari atribut ke-*v* dari pusat cluster ke-*j*

w_{ij} : Nilai bobot gabungan dari x_i dengan c_j

X : Matrix data dengan ukuran $m \times n$

C : Matrix pusat cluster dengan ukuran $g \times n$

W : Matrix bobot dengan ukuran $m \times g$

Matriks pheromone merupakan representasi jejak untuk setiap agen semut. Matriks pheromone dinormalisasikan menggunakan persamaan berikut (Shelokar dkk, 2006):

$$P_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^{g} \tau_{ik}}, j = 1,...,g \quad \ldots (3)$$

P_{ij} merupakan matriks probabilitas normalisasi pheromone untuk elemen *i* terhadap cluster *j*.

Jarak antara obyek *i* dan *cluster j* dari semut *k* ($d_k(i,j)$) dapat didefinisikan pada persamaan berikut (Kao dan Cheng, 2006):

$$d^k(i,j) = \sqrt{\sum_{v=1}^{n}(x_{iv} - c_{jv}^k)^2} \quad \ldots (4)$$

Pemilihan cluster *j* oleh setiap semut menggunakan dua strategi, yaitu eksploitasi dan eksplorasi. Adapun persamaan untuk melakukan eksploitasi adalah sebagai berikut (Dorigo, 1999):

$$j = \begin{cases} \arg\max_{u \in N_i} \{[\tau(i,u)]^{\alpha} [\eta^k(i,u)]^{\beta}\} & \text{if} \quad q \le q_0 \\ P^k(i,j) & \text{otherwise} \end{cases} \quad \ldots (5)$$

Dan persamaan eksplorasi sebagai berikut (Dorigo dkk, 1999):

$$P^k(i,j) = \frac{[\tau(i,u)]^{\alpha} [\eta^k(i,u)]^{\beta}}{\sum_{j=1}^{g} [\tau(i,u)]^{\alpha} [\eta^k(i,u)]^{\beta}} \quad \ldots (6)$$

Di mana nilai η_{ij}^k, didapat dari persamaan berikut (Dorigo dkk, 1999):

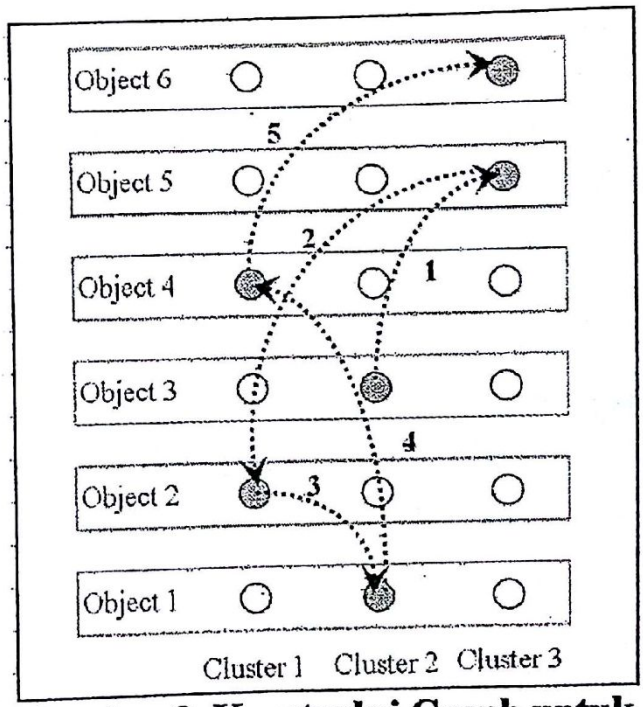
$$\eta_{ij}^k = \frac{1}{d^k(i,j)} \quad .$$

Untuk mengupdate nilai pheromone digunakan persamaan berikut (Dorigo dkk, 1999):

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \quad \ldots (7)$$

Di mana $\Delta\tau_{ij}^h = \frac{1}{J^h}$, J^h merupakan nilai fungsi obyektif, nilai $\alpha \ge 0$ dan nilai $\beta > 0$.

Dalam algoritma *Ant Colony Optimization Clustering (ACOC)*, ruang solusi dimodelkan sebagai suatu graph dengan matrik node obyek-cluster. Jumlah baris sama dengan *m*, dan jumlah kolom sama dengan *g*. Setiap node diwakilkan dengan N(i,j) yang artinya bahwa obyek data *i* ditentukan ke cluster *j*. Setiap semut dapat menempati hanya satu dari *g* node untuk setiap obyek.



Gambar 2. Konstruksi Graph untuk ACOC (Kao dan Cheng, 2006)

Pada gambar 2, mengilustrasikan suatu contoh dari konstruksi graph untuk permasalahan clustering, di mana lingkaran kosong menandakan node-node yang tidak terkunjungi dan lingkaran penuh menandakan node-node terkunjungi oleh semut-semut. Berdasarkan hasil clustering pada gambar 2, solution string yang terbentuk adalah (2,1,2,1,3,3).

Pada graph, setiap semut bergerak dari satu node ke node yang lainnya, dan meninggalkan pheromone pada node dan membentuk suatu solusi pada setiap langkah jalurnya. Pada tiap langkahnya, setiap semut secara acak memilih obyek yang belum memiliki kelompok dan menambahkan suatu node yang baru ke sebagian solusinya berdasarkan kedua informasi intensitas pheromone dan heuristic.

Dalam ACOC, semut-semut meninggalkan pheromone pada node-node. Node-node dengan pheromone yang tinggi akan lebih atraktif pada semut. ACOC menggunakan sebuah Matriks Pheromone untuk menyimpan nilai-nilai pheromone. Informasi heuristic mengindikasikan keinginan menentukan suatu obyek data pada suatu bagian cluster. Hal ini mewajibkan untuk menghitung Euclidean distance antara obyek data yang tercluster dengan setiap pusat cluster dari beberapa semut. Node-node dengan nilai heuristic yang lebih tinggi akan dipilih oleh semut-semut. Setiap semut akan membawa sebuah matrik pusat cluster (C^k) untuk menyimpan pusat clusternya dan mengubah nilainya setiap langkah clustering.

Tabel 1. Prosedur Lengkap Dari ACOC

Langkah	Keterangan
1	Melakukan inisialisasi semua semut. Mulai iterasi baru sampai jumlah semut. Inisialisasi matriks *Pheromone* untuk setiap semut. Elemen-elemen dari matriks pheromone ditentukan agar terpilih nilai yang kecil (τ_0).
2	Lakukan normalisasi matriks pheromone menggunakan persamaan (3).
3	Inisialisasi Solution String secara acak untuk setiap semut. Hitung bobot matrik (W^k) untuk tiap semut, dan Hitung Matriks pusat cluster (C^k) menggunakan persamaan (2) dan, di mana $k=1..R$. R adalah jumlah semut, $R \leq m$.
4	Lakukan langkah 2 dan 3 sampai iterasi mencapai jumlah semut.
5	Memulai iterasi baru. Hitung matriks jarak antara Matriks Data dengan Matriks Pusat Cluster menggunakan persamaan (4).
6	Menghitung pemilihan cluster j, untuk menentukan j bagi i yang terpilih, ada dua strategi yang digunakan yaitu eksploitasi dan eksplorasi. Bangkitkan suatu bilangan acak q. jika $q < q_0$ maka dilakukan perhitungan eksploitasi menggunakan persamaan (5). Jika tidak, maka dilakukan perhitungan eksplorasi menggunakan persamaan (6).
7	Bentuk Solution String dari hasil pemilihan cluster. Buat Matriks bobot (W^k) untuk setiap semut. Perbaiki Matriks pusat cluster (C^k).
8	Hitung fungsi Obyektif dari setiap semut menggunakan persamaan (1). Setelah itu urutkan secara ascending semua nilai fungsi obyektif dari semua semut. Solution string berdasarkan nilai fungsi obyektif tertinggi digunakan sebagai solution string terbaik.

Langkah	Keterangan
9	Lakukan update matriks pheromone menggunakan persamaan (7). di mana ρ merupakan *pheromone evaporation rate* yang nilainya berkisar antara 0 dan 1 $(0.0 < \rho < 1.0)$.
10	Lakukan langkah 5 sampai 9, jika jumlah iterasi mencapai maksimum iterasi yang ditentukan maka proses clustering berhenti, kemudian ambil solution string berdasarkan fungsi obyektif terbaik.

2.4 Analisa Klaster *(Cluster Analysis)*

Analisa cluster adalah suatu teknik analisa *multivariate* (banyak variabel) untuk mencari dan mengorganisir informasi tentang variabel tersebut sehingga secara relatif dapat dikelompokkan dalam bentuk yang homogen dalam sebuah klaster (Nadler dan Smith, 1993). Analisis cluster diukur dengan menggunakan nilai *variance* atau *error ratio*. *Variance* digunakan untuk mengukur nilai penyebaran dari data-data hasil clustering dan dipakai untuk data bertipe *unsupervised*. Secara umum, bisa dikatakan sebagai proses menganalisa baik tidaknya suatu proses pembentukan *cluster*. Analisa cluster bisa diperoleh dari kepadatan cluster yang dibentuk (*cluster density*). Kepadatan suatu cluster bisa ditentukan dengan *variance within cluster* (V_w) dan *variance between clusters* (V_b). Varian tiap tahap pembentukan cluster bisa dihitung dengan persamaan (8) berikut (Nadler dan Smith, 1993):

$$Vc^2 = \frac{1}{n_c - 1}\sum_{i=1}^{n}(y_i - \overline{y_c})^2 \quad \ldots (8)$$

Dimana:

Vc^2 : varian pada cluster c, $c = 1..k$, dimana k = jumlah cluster

n_c : jumlah data pada cluster c

y_i : data ke-i pada suatu cluster

$\overline{y_i}$: rata-rata dari data pada suatu cluster

Selanjutnya dari nilai varian diatas, kita bisa menghitung nilai *variance within cluster* (V_w) dengan persamaan (9) berikut (Nadler dan Smith, 1993):

$$Vw = \frac{1}{N - c}\sum_{i=1}^{c}(n_i - 1).Vi^2 \quad \ldots (9)$$

Dimana:

N : Jumlah semua data

n_i : Jumlah data cluster i

V_i : Varian pada cluster i

Dan nilai *variance between cluster* (V_b) dengan persamaan (10) berikut (Nadler dan Smith, 1993):

$$Vb = \frac{1}{c-1}\sum_{i=1}^{c} n_i(\overline{y_i} - \overline{y})^2 \quad \ldots (10)$$

Di mana, $\overline{y}$ adalah rata-rata dari $\overline{y_i}$.

Salah satu metode yang digunakan untuk menentukan cluster yang ideal adalah batasan *variance*, yaitu dengan menghitung kepadatan *cluster* berupa *variance within cluster* (V_w) dan *variance between cluster* (V_b) (Veenman dkk, 2002). Cluster yang ideal mempunyai V_w minimum yang merepresentasikan *internal homogenity* dan maksimum V_b yang menyatakan *external homogenity*. Cluster disebut ideal jika memiliki nilai V_w seminimal mungkin dan V_b semaksimal mungkin. Nilai *variance* (V) dapat dihitung menggunakan persamaan (11).

$$V = \frac{Vw}{Vb} \quad \ldots (11)$$

Meskipun minimum V_w menunjukkan nilai cluster yang ideal, tetapi pada beberapa kasus kita tidak bisa menggunakannya secara langsung untuk mencapai global optimum. Jika dipaksakan, maka solusi yang dihasilkan akan jatuh pada local optima.

Renals (2009) menyatakan bahwa metode lainnya untuk menganalisis hasil klaster adalah dengan menghitung *Sum Squared Error (SSE)*. Untuk setiap *data point*, nilai kesalahan didapatkan dari perhitungan jarak dengan klaster terdekatnya. Untuk mendapatkan nilai SSE, nilai error yang dikuadratkan kemudian dijumlahkan semua. Perhitungan jarak digunakan persamaan *squared Euclidean distance*. (Gose, dkk, 1996 dan Renals, 2009). Untuk mendapatkan nilai SSE digunakan persamaan (12).

$$SSE = \sum_{i=1}^{K}\sum_{x \in C_i} dist^2(m_i, x) \quad \ldots (12)$$

Dimana x adalah *data point* dalam klaster C_i dan m_i merupakan *point representative* untuk klaster C_i (pusat klaster).

Salah satu cara untuk mereduksi *SSE* adalah dengan meningkatkan nilai K (jumlah klaster). Hasil klastering yang baik yaitu memiliki nilai *SSE* dengan *error* terkecil. Klastering yang baik dengan K yang lebih kecil memiliki nilai *SSE* yang rendah daripada klastering dengan K yang besar.

3. METODOLOGI PENELITIAN

3.1 Objek Penelitian

Dalam penelitian ini yang dijadikan objek penelitian adalah kumpulan Dataset yang dapat diunduh dari internet dengan alamat http://www.cs.sfu.ca/~wangk/ucidata/dataset/.

3.2 Metode Pengumpulan Data

Didalam mengumpulkan data-data penulis menggunakan metode: 1) Dokumentasi, yaitu mengambil dataset yang dibutuhkan sebagai data pengujian, 2) Studi Literatur, yaitu mempelajari semua literatur yang berkaitan dengan klastering menggunakan *Ant Colony Optimization*.

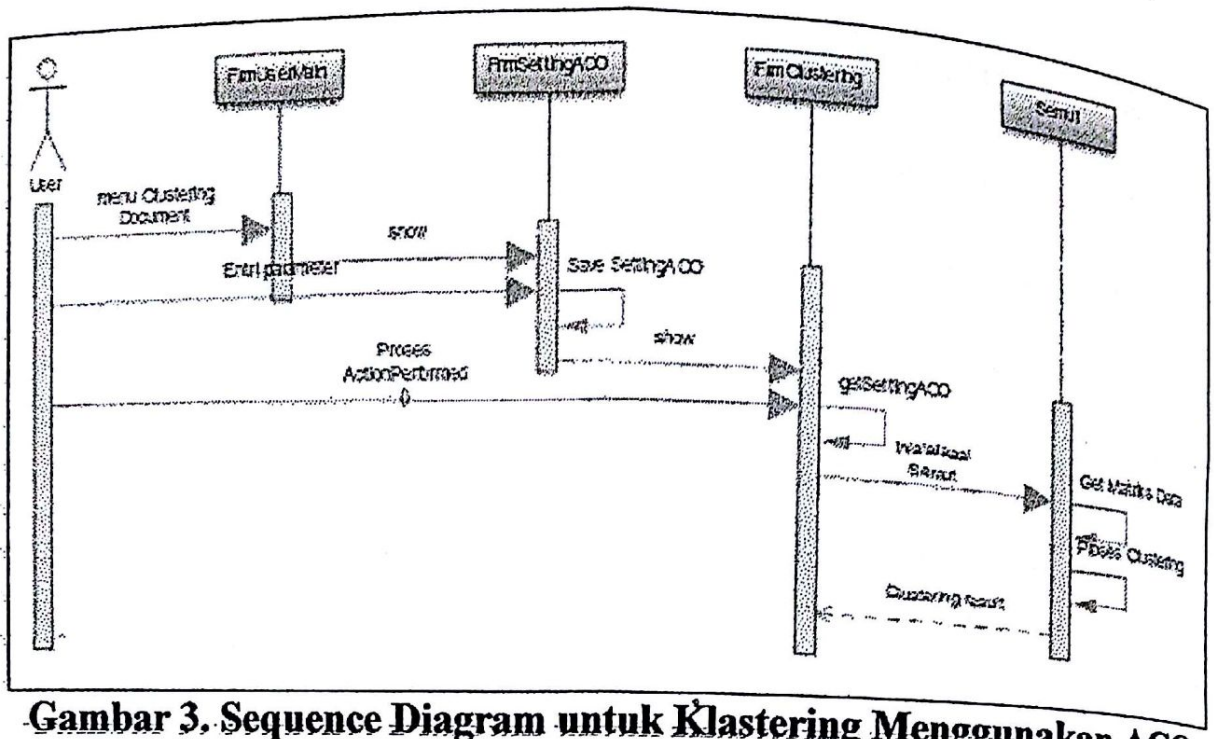
3.3 Metode Pengembangan Perangkat Lunak

Penelitian ini menggunakan metode pengembangan perangkat lunak yang terdiri dari tahap-tahap sebagai berikut: 1) Analisa, Merupakan tahap untuk menganalisa data yang diperoleh sebagai bahan pengujian, 2) Perancangan. Merupakan tahap perancangan terhadap aplikasi klastering, 3) Implementasi. Merupakan tahap untuk mengimplementasikan hasil rancangan sistem menjadi sebuah perangkat lunak, 4) Pengujian. Merupakan tahap uji coba dari perangkat lunak yang telah diimplementasikan, dan 5) Evaluasi dan Perbaikan Kesalahan. Merupakan tahap untuk melakukan evaluasi dan dan perbaikan terhadap kesalahan-kesalahan yang terjadi dalam perangkat lunak yang dibuat.

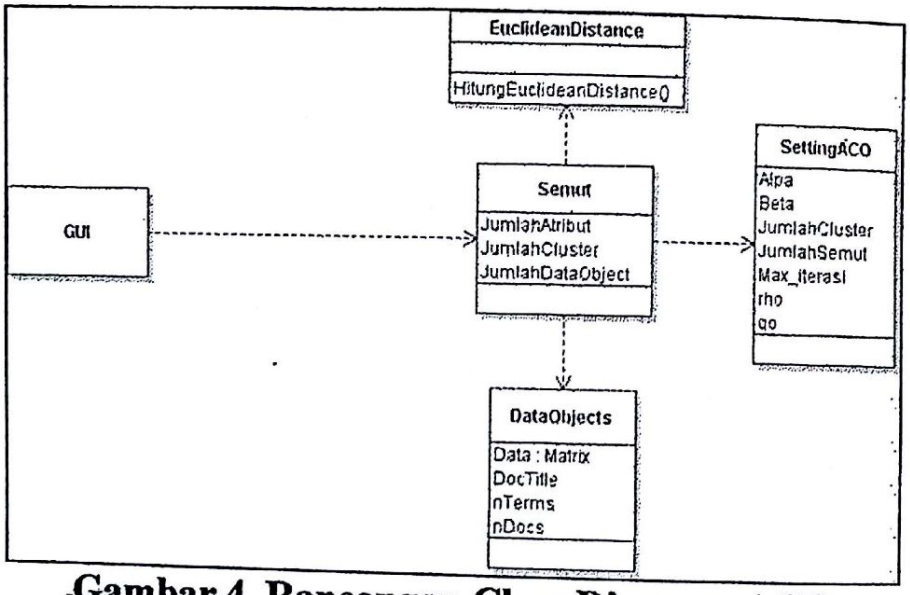
4. PEMBAHASAN

4.1 Rancangan Sequence Diagram dan Class Diagram

Dalam klastering menggunakan algoritma Ant Colony Optimization, class Semut dirancang agar memiliki hubungan *one-to-many*. Class semut bergantung pada Class SettingACO di mana sejumlah parameter tersebut yang digunakan untuk klastering. Class Semut mengambil data objects dari class DataObjects. Untuk perhitungan euclidean distance digunakan class EuclideanDistance. Rancangan class diagram ACO dapat dilihat pada gambar 3.



Gambar 3. Sequence Diagram untuk Klastering Menggunakan ACO



Gambar 4. Rancangan Class Diagram ACO

4.2 Implementasi Algoritma ACO

Implementasi klastering dapat dilihat pada pseudocode berikut:

```
/* Inisialisasi Semua Semut */
For i=1 to Jumlah_Semut do
      Buat Matrik Pheromone untuk setiap Semut ke-i
      Hitung Normalisasi Matrik Pheromone
      Buat Solution String i untuk setiap semut
      Hitung Matriks bobot
      Hitung matriks pusat cluster
End For
```

```
/* Proses Clustering */
For i=1 to Max_Iterasi do
    For j=1 to Jumlah_Semut do
        Hitung matriks jarak antara matriks data dengan matriks pusat
cluster
        Hitung matriks η_ij^k
        Buat angka acak (q)
        If q ≤ q0 then
            j = { arg max_{u∈N_i} {[τ(i,u)]^α [η^k(i,u)]^β}   if q ≤ q0
                { P^k(i,j)                                    otherwise
        Else
            P^k(i,j) = [τ(i,u)]^α [η^k(i,u)]^β / Σ_{j=1}^{g} [τ(i,u)]^α [η^k(i,u)]^β
        End If
        Bentuk Solution String
        Hitung Matriks bobot
        Perbaiki Matrik Pusat Cluster
        Hitung Fungsi Objectif
        Update Matriks Pheromone
        Hitung Normalisasi Matrik Pheromone
    End For
End For
```

Gambar 5. Pseudocode Klastering Dokumen

Dalam implementasi klastering dokumen, diperlukan beberapa variable. Parameter yang di inisialisasikan adalah: 1) Jumlah klaster, 2) Nilai α yaitu tetapan pengendali intensitas jejak semut, nilai $\alpha \geq 0$, 3) Nilai β yaitu tetapan pengendali visibilitas, nilai $\beta \geq 0$, 4) Banyak agen semut, 5) Nilai ρ yaitu tetapan penguapan jejak semut, nilai ρ harus $0 < \rho < 1$ untuk mencegah jejak pheromone yang tak terhingga, 6) Iterasi_max yaitu jumlah siklus maksimum, dan 7) Nilai $q0$, yaitu nilai probabilitas turnable parameter, nilai $0 < q_0 < 1$.

4.3 Pengujian

Pengujian meliputi pengujian nilai *variance*, pengujian nilai *Sum Squared Error*, dan pengujian kinerja aplikasi berdasarkan waktu pemrosesan. Semua eksperiman dilakukan pada sebuah Notebook HP Compaq, INTEL Core 2 DUO T5300, 1.73 GHz, Memory 2 GB dan Sistem Operasi Windows Vista. Tabel 2 menunjukkan sejumlah dataset multidimensional yang digunakan untuk pengujian.

Tabel 2. Tabel Dataset

No	Dataset	Number of Instances	Number of Attributes
1	DNA	2000	180
2	Waveform	5000	21
3	Page-blocks	5473	10
4	Ann-thyroid	7200	21
5	Letter-recognition	20000	16
6	Shuttle	43500	9

Pengujian klastering dataset tersebut menggunakan algoritma *Ant Colony Optimization* menggunakan parameter sebagai berikut : jumlah klaster (K) = 10, jumlah iterasi = 10, $\alpha = 1$, $\beta = 2$, jumlah semut = 30, $\rho = 0.1$, $\tau_0 = 0.1$ dan $q_0 = 0.01$. Tabel hasil pengujian sejumlah dataset dapat dilihat pada tabel 3 dan tabel 4 merupakan hasil pengujian nilai *variance*.

Tabel 3. Hasil Pengujian Dataset

No	Dataset	Waktu	Variance	SSE
1	DNA	15m:23s	0,02541473	0,15941751
2	Waveform	5m:16s	0,115509465	0,105100766
3	Page-blocks	3m:16s	0,024962544	0,76241654
No	**Dataset**	**Waktu**	**Variance**	**SSE**
4	Ann-thyroid	7m:33s	0,025718259	0,4756689
5	Letter-recognition	17m:31s	0,038778864	1,1945972
6	Shuttle	25m:4s	0,006308183	0,58437836

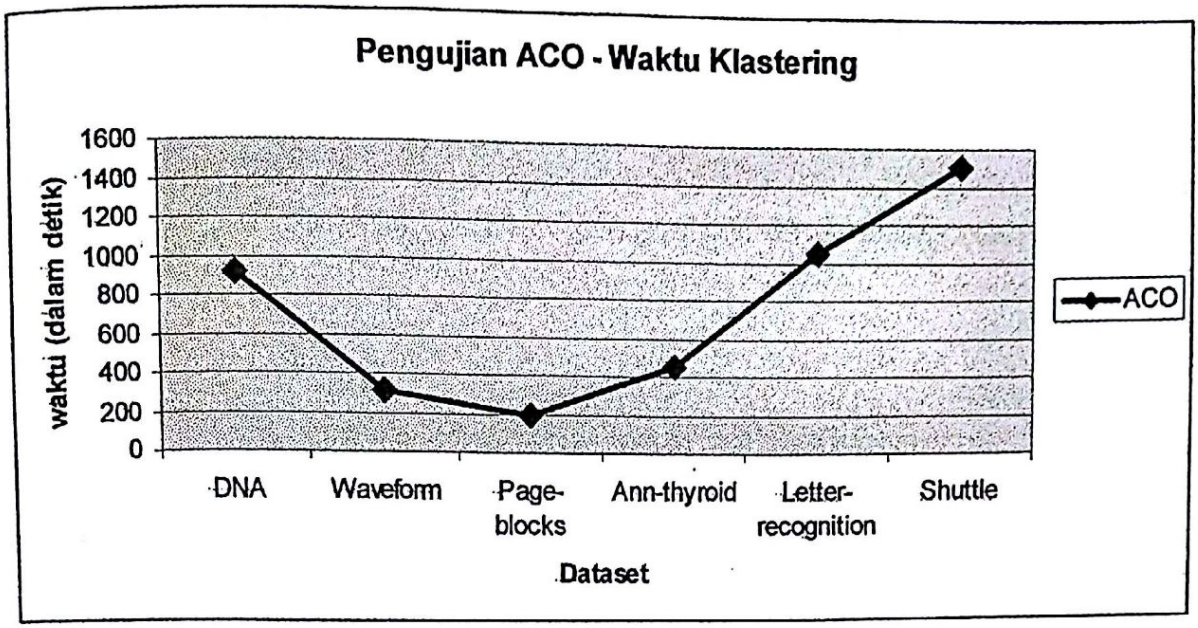
Keterangan: m = menit (*minute*); s = detik (*second*)

Tabel 4. Hasil Pengujian Nilai *Variance*

No	Dataset	Vw	Vb	V
1	DNA	1,07E-04	4,20E-03	0,02541473
2	waveform	2,41E-04	0,002087924	0,115509465
3	Page-blocks	4,35E-04	0,017441763	0,024962544
4	ann-thyroid	5,02E-04	0,019528711	0,025718259
5	letter-recognition	3,52E-05	9,08E-04	0,038778864
6	shuttle	4,79E-05	0,007588219	0,006308183

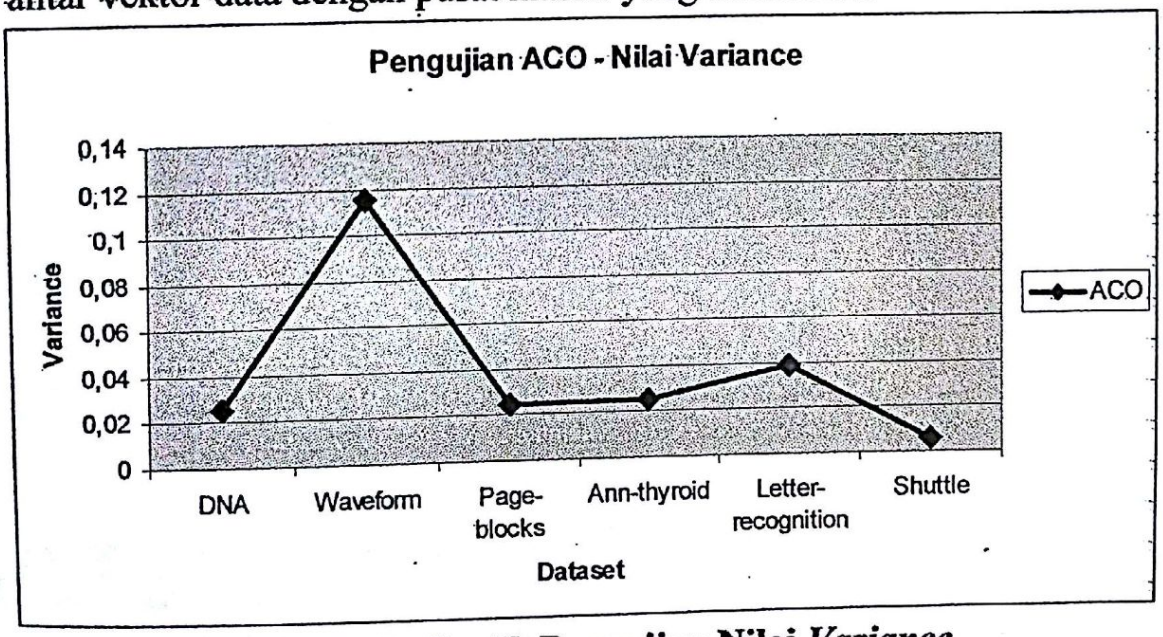
Keterangan: *variance within cluster* (V_w), *variance between cluster* (V_b) dan *variance*(V).

Berdasarkan tabel 3 dapat digambarkan grafik pengujian kinerja sistem berdasarkan waktu klasteringnya. Gambar 6 menunjukkan grafik hasil klastering dengan waktu klastering. Dari grafik tersebut dapat disimpulkan bahwa algoritma Ant Colony Optimization akan memerlukan waktu yang relative lama seiring dengan ukuran multidimensional dari dataset.



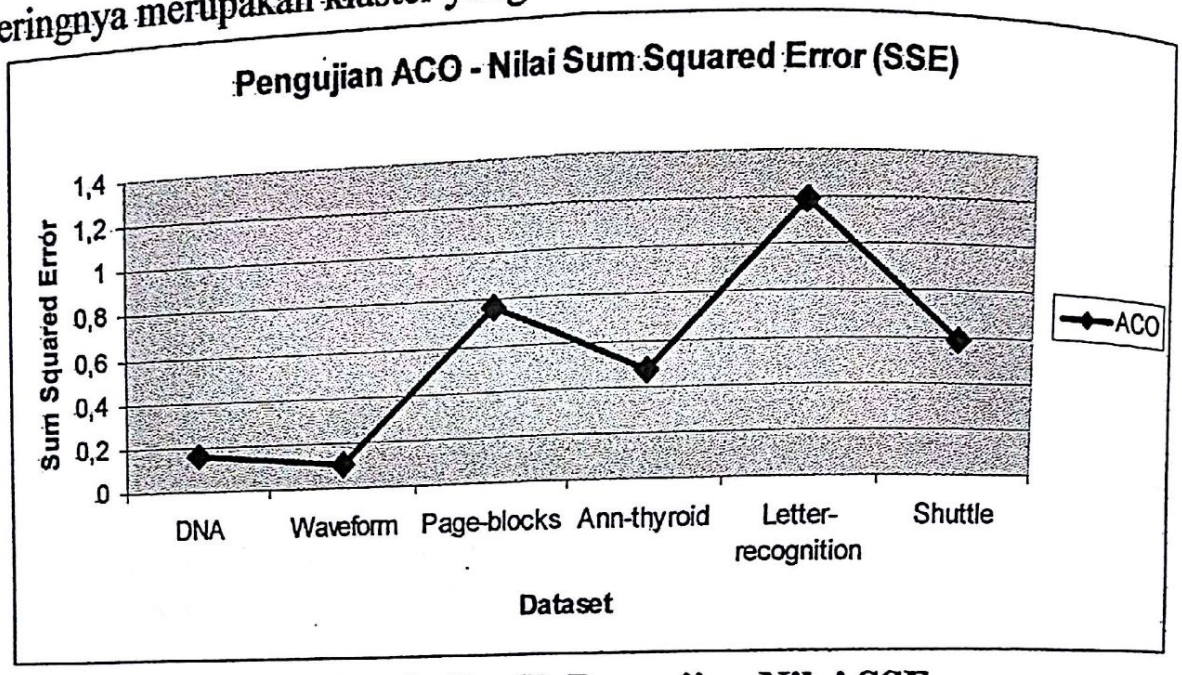
Gambar 6. Grafik Pengujian Waktu Klastering

Berdasarkan tabel 3 dan 4 dapat digambarkan grafik pengujian nilai *variance*. Gambar 7 menunjukkan grafik hasil klastering dengan nilai *variance*. Dari gambar grafik tersebut dapat disimpulkan bahwa algoritma *Ant Colony Optimization* memiliki nilai *variance* yang rendah dan dapat dikatakan hasil klasteringnya merupakan klaster yang ideal. Hal ini dikarenakan hasil klastering menghasilkan *intraclass similarity* (kesamaan di dalam klas) yang tinggi dan *interclass similarity* (kesamaan antar klas) yang rendah, hal ini didasarkan pada jarak antar vektor data dengan pusat klaster yang dihasilkan.



Gambar 7. Grafik Pengujian Nilai *Variance*

Berdasarkan tabel 3 dapat digambarkan grafik pengujian nilai *Sum Squared Error* (SSE). Gambar 5 menunjukkan grafik hasil klastering dengan nilai *SSE*. Dari gambar grafik tersebut dapat disimpulkan bahwa algoritma *Ant Colony Optimization* memiliki nilai *SSE* yang rendah dan dapat dikatakan hasil klasteringnya merupakan klaster yang baik.



Gambar 8. Grafik Pengujian Nilai SSE

5. SIMPULAN

Dari hasil analisis, rancangan, penelitian dan pembahasan dari sejumlah bahasan di atas, maka penulis dapat menarik sejumlah simpulan yang cukup berarti, sebagai berikut:

1) Klastering menggunakan Algoritma *Ant Colony Optimization* memerlukan waktu pemrosesan yang sangat lama, hal ini dipengaruhi oleh besarnya ukuran dataset multidimensional serta banyaknya jumlah agen semut dan jumlah iterasi yang diberikan terhadapnya.
2) Hasil klastering menghasilkan *intraclass similarity* (kesamaan di dalam klas) yang tinggi dan *interclass similarity* (kesamaan antar klas) yang rendah, hal ini didasarkan pada jarak antar vektor data dengan pusat klaster yang dihasilkan.
3) Hasil klastering menggunakan *Ant Colony Optimization* merupakan hasil klaster yang baik hal ini dapat dilihat dari perolehan nilai *Sum Squared Error* dari sejumlah pengujian multidimensional dataset.
4) Untuk penelitian lebih lanjut, topik berikut dapat menjadi bahan pertimbangan: a) disarankan pengembangan aplikasi ini selanjutnya

menggunakan teknik pemrograman dan struktur data yang bisa menangani komputasi matriks yang besar dan lebih cepat dalam memproses klastering multidimensional dataset, dan b) dapat mengembangkan dan memodifikasi algoritma *Ant Colony Optimization* menjadi algoritma yang lebih handal untuk memberikan solusi yang lebih optimal.

DAFTAR RUJUKAN

Dorigo, M.; Bonabeau, E.; Theraulaz, G. 1999. *Swarm Intelligence From Natural to Artificial Systems*. Oxford University Press. New York. USA.

Gose, E.; Johnsonbaugh, R.; Jost, S. 1996. *Pattern recognition and Image Analysis*. Prentice Hall, USA.

Kao, Y.; Cheng, K. 2006. An ACO-Based Clustering Algorithm, *Ant Colony Optimization and Swarm Intelligence*, 4150:340-347. Springer Berlin/Heidelberg.

Kennedy, J.; Eberchart, R.C.; Shi, Y. 2001. *Swarm Intelligence*. Morgan Kaufmann Publisher. USA.

Nadler, M.; Smith, E.P. 1993. *Pattern Recognition Engineering*. John Wiley & Sons.,Inc. USA.

Renals, S. 2009. *Clustering*. Learning and Data Note 3 (v2.2).

Shelokar, Jayaraman, V.K.; Kulkarni, B.D. 2004. An ant colony approach for clustering, *Analytica Chimica Acta*, 509(2):187-195.

Turban, E.; Aronson, J. 1998. *Decision Support Systems and Intelligent Systems. Fifth Edition*. Prentice-Hall, Inc.

Veenman, C.J.; Reinders, M.J.T.; Backer, E. 2002. A Maximum Variance Cluster Algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273-1280.