# ANT COLONY OPTIMIZATION UNTUK CLUSTERING DOKUMEN HASIL PENCARIAN

DAVID

Program Studi Teknik Informatika
Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jln. Merdeka No. 372 Pontianak, Kalimantan Barat
David_Liauw@yahoo.com dan David_Liauw@stmikpontianak.ac.id

ABSTRAK

Clustering dokumen merupakan salah satu topik penelitian yang popular dalam data mining. Pada penelitian ini membuat sistem aplikasi Clustering dokumen hasil pencarian menggunakan algoritma clustering *Ant Colony Optimization*. Aplikasi yang dibangun dapat digunakan untuk mengelompokkan dokumen hasil pencarian dan memudahkan pencarian dokumen. Dokumen yang di-clustering-kan hanya untuk artikel pada jurnal, tesis, proposal tesis, ebook dan dokumen lainnya. *Indexing* dan *searching* dokumen menggunakan Lucene sebagai *search engine*. Hasil penelitian adalah pengujian waktu proses clustering, nilai rasio *variance* dan nilai *Sum Squared Error*. Eksperimen dilakukan terhadap kumpulan dokumen. Dalam penelitian ini disimpulkan bahwa secara keseluruhan hasil clustering dari algoritma *Ant Colony Optimization* memiliki nilai rasio *variance* yang minimum dan masing-masing hasil clustering menghasilkan *intraclass similarity* yang tinggi dan *interclass similarity* yang rendah.

Kata Kunci : Clustering Dokumen, *Ant Colony Optimization*, *Sum Squared Error, rasio variance*

## 1. Pendahuluan

Salah satu penerapan teknik data mining clustering untuk mengcluster dokumen. Dokumen yang dicluster akan membentuk *cluster-cluster* yang memudahkan pengelompokkan dokumen. Pencarian dokumen yang tanpa *clustering* akan menampilkan semua dokumen yang mengandung *keyword* tertentu. Contohnya seperti pada search engine-nya *Google* atau *Yahoo*.

Dokumen-dokumen pada perpustakaan seperti artikel pada jurnal, tesis, buku, proposal tesis dan lain-lainnya, biasanya tidak dikelompokkan berdasarkan isinya. Sehingga akan sulit untuk melakukan pencarian dokumen berdasarkan isinya. Untuk itu diperlukan aplikasi yang dapat mengelompokkan dan memudahkan pencarian dokumen. Banyak sekali metode clustering yang dapat diimplementasikan, namun penulis menggunakan algoritma *Ant Colony Optimization*. Hal ini dikarenakan masih kurangnya penelitian clustering dokumen menggunakan algoritma *Ant Colony Optimization* sebagai metode *partitioning clustering*. Dokumen yang diclusteringkan hanya untuk artikel pada jurnal, tesis, proposal tesis dan ebook.

Tujuan penelitian ini adalah membuat aplikasi clustering dokumen hasil pencarian menggunakan algoritma clustering *Ant Colony Optimization* serta menguji algoritma tersebut dari segi waktu proses clustering, nilai rasio *variance* dan nilai *Sum Squared Error*.

## 2. Metode Penelitian

Metode Penelitian meliputi tahapan analisis ini dilakukan pada saat tahap perencanaan telah selesai. Pada tahapan ini melakukan penelitian lanjutan diperlukan untuk memperoleh data yang lebih terperinci, yang bertujuan untuk keperluan pengembangan sistem secara teknis. Langkah-langkah yang perlu dilakukan dalam tahapan analisis sistem ini adalah sebagai berikut : a) Menganalisis kebutuhan sistem (requirements analysis), dalam hal ini dilakukan analisis mengenai sistem clustering dokemen yang dibutuhkan; b) Menganalisis hasil penelitian untuk menentukan pilihan perancangan (decision analysis), dalam hal ini dilakukan analisis mengenai perancangan yang akan digunakan untuk system clustering yang akan dibuat.

Selanjutnya tahapan perancangan sistem adalah tindak lanjut dari analisis system, tahapan yang dilakukan untuk perancangan sistem aplikasi adalah : a) Mengidentifikasi kebutuhan informasi clustering dokumen; b) Menentukan variabel input sistema; c) Menentukan proses clustering pada sistema; d) Menyusun diagram UML yang mempunyai fungsi membuat model perancangan sistem dan proses dalam simbol-simbol tertentu; e) Menyusun prototype sistem aplikasi baik input maupun output.

Setelah itu dilanjutkan pada tahap Implementasi Sistem dimana pada tahap ini dilakukan pembahasan clustering dokumen menggunakan algoritma Ant Colony

Optimization serta diimplementasikan ke dalam bahasa pemrograman java.

3. Tinjauan Pustaka

Ant Colony Optimization

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 [3].

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

Di mana $\left\|X_i - C_j\right\| = \sqrt{\sum_{v=1}^{n}(x_{iv} - c_{jv})^2}$ 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 $w_{ij} = 0$.

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

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 $m \times g$

Jejak untuk setiap agen semut direpresentasikan ke dalam matriks *pheromone*. Matriks pheromone dinormalisasikan menggunakan persamaan berikut [2]:

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

$P_{ij}$ 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):

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

Pemilihan cluster *j* oleh setiap semut menggunakan dua strategi, yaitu eksploitasi dan eksplorasi. Adapun persamaan untuk melakukan eksploitasi adalah sebagai berikut [1]:

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

Dan persamaan eksplorasi sebagai berikut [1]:

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

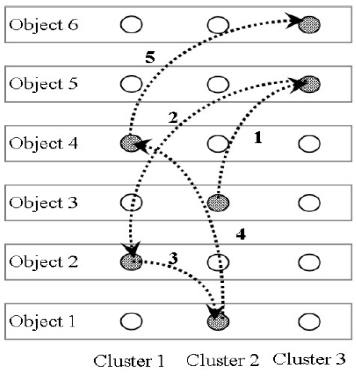
Dimana nilai $\eta_{ij}^k$, didapat dari persamaan berikut [1]: $\eta_{ij}^k = \frac{1}{d^k(i,j)}$

Untuk mengupdate nilai pheromone digunakan persamaan berikut [1]:

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

Dimana $\Delta\tau_{ij}^h = \frac{1}{J^h}$, $J^h$ 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. Pada gambar 1, 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 1, solution string yang terbentuk adalah (2,1,2,1,3,3).




Gambar 1. Konstruksi Graph untuk ACOC [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*.

Prosedur lengkap dari *ACOC* dideskripsikan sebagai berikut:

Langkah 1 : Melakukan inisialisasi semua semut. Mulai iterasi baru sampai jumlah semut. Inisialisasi matriks *Pheromone* untuk setiap semut. Elemen-elemen dari matriks pheromone ditentukan dengan nilai yang kecil ($\tau_0$).

Langkah 2 : Lakukan normalisasi matriks pheromone menggunakan persamaan (3).

Langkah 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, $R \leq m$.

Langkah 4 : Lakukan langkah 2 dan 3 sampai iterasi mencapai jumlah semut.

Langkah 5 : Memulai iterasi baru. Hitung matriks jarak antara Matriks Data dengan Matriks Pusat Cluster menggunakan persamaan (4).

Langkah 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).

Langkah 7 : Bentuk Solution String dari hasil pemilihan cluster. Buat Matriks bobot ($W^k$) untuk setiap semut. Perbaiki Matriks pusat cluster ($C^k$).

Langkah 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 9 : Lakukan update matriks pheromone menggunakan persamaan (7). dimana $\rho$ merupakan *pheromone evaporation rate* yang nilainya berkisar antara 0 dan 1 ($0.0 < \rho < 1.0$) .

Langkah 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.

Analisa Clustering

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 cluster [4]. 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 cluster* ($V_b$). Varian tiap tahap pembentukan cluster bisa dihitung dengan persamaan (8) berikut [4]:

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

Di mana,

$Vc^2$ : 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 [4]:

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

Di mana,

$N$ : 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 [4]:

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

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* ($V_b$). 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).

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

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 cluster adalah dengan menghitung *Sum Squared Error (SSE)*[5]. Untuk setiap *data point*, nilai kesalahan didapatkan dari perhitungan jarak dengan cluster terdekatnya. Untuk mendapatkan nilai SSE, nilai error yang dikuadratkan kemudian dijumlahkan semua. Perhitungan jarak digunakan persamaan *squared Euclidean distance* [6],[5]. Untuk mendapatkan nilai SSE digunakan persamaan (12).

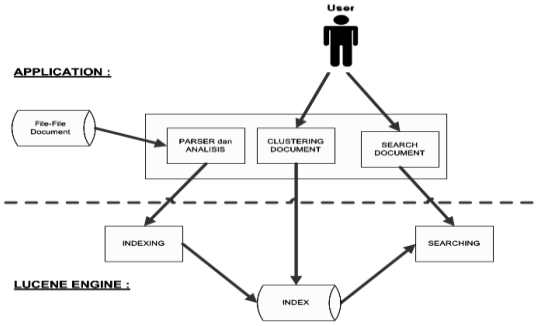
$$SSE = \sum_{i=1}^{K}\sum_{x \in C_i} dist^2(m_i, x) \quad \text{..... (12)}$$

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

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

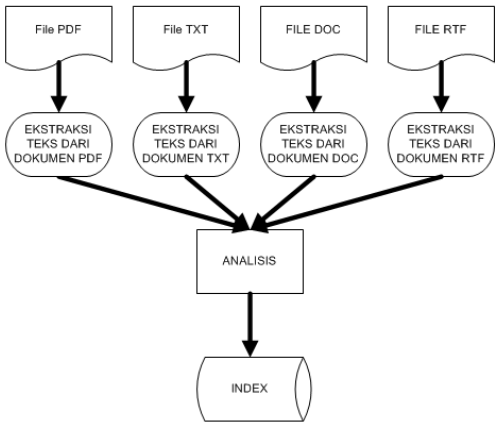
4. Hasil Penelitian dan Pembahasan

Aplikasi yang dibangun dalam penelitian ini adalah aplikasi pencarian untuk koleksi dokumen perpustakaan. Dalam perancangan aplikasi pada *digital library* ini, menggunakan library *Apache Lucene* untuk melakukan *indexing* dan *searching*. Selanjutnya untuk melakukan *clustering* dokumen menggunakan algoritma *ant colony optimization*. Arsitektur aplikasi yang dibangun dapat dilihat pada gambar 2.




Gambar 2. Arsitektur Aplikasi DocClus

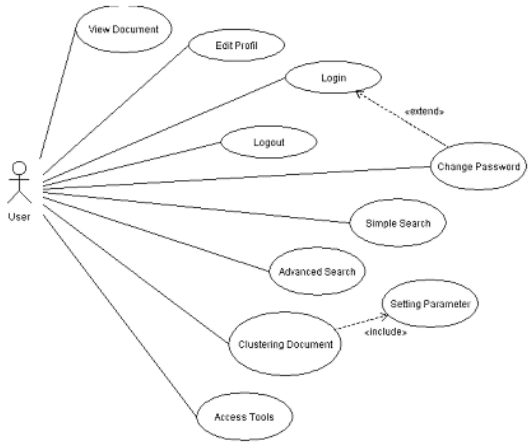
Dalam arsitektur aplikasi DocClus terdapat dua bagian, yaitu bagian *Clustering* dan *Searching*. File-file dokumen seperti file tesis, proposal tesis, artikel jurnal, ebook dan dokumen lainnya terlebih dahulu melalui proses ekstraksi teks (*Parser*) untuk kemudian dianalisis menggunakan *library lucene*, seperti yang terlihat pada gambar 3. Tahapan analisis meliputi memproses stopword, analisa standar, lower case dan porter stemming. Hasil analisis kemudian diindex dan tersimpan dalam file index. Proses searching dibedakan menjadi dua, yaitu Simple Search dan *Advanced Search*. *Simple Search* menggunakan *query Db4o* untuk melakukan pencarian sedangkan *Advanced Search* menggunakan pencarian term pada index menggunakan *library lucene*. Proses clustering menggunakan matriks term frekuensi dari index sebagai matriks data.




Gambar 3. Indexing Menggunakan Lucene

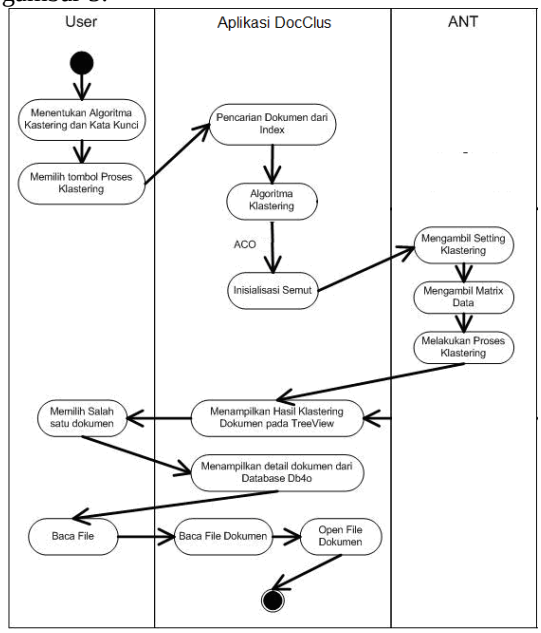
Gambar 4 menunjukkan use case diagram untuk actor user. Aplikasi digital library ini memiliki

fitur indexing, simple search, advanced search serta clustering dokumen.

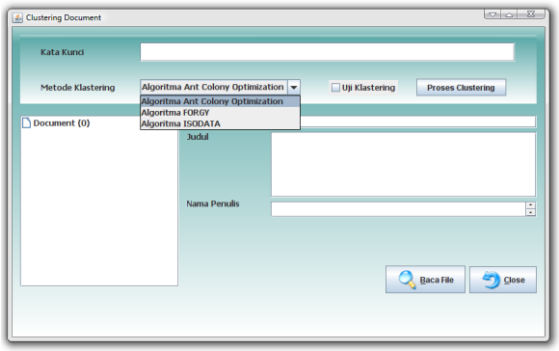



Gambar 4. Use Case dari actor user

Setelah kata kunci dan algoritma klastering ditentukan, maka proses klastering akan dilakukan dengan terlebih dahulu melakukan pencarian pada index dan mengambil matriks data. Berikut activity diagram dapat dilihat pada gambar 5.

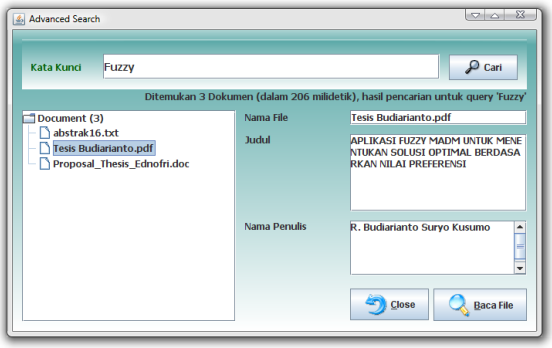



Gambar 5. Activity Diagram Clustering Dokumen



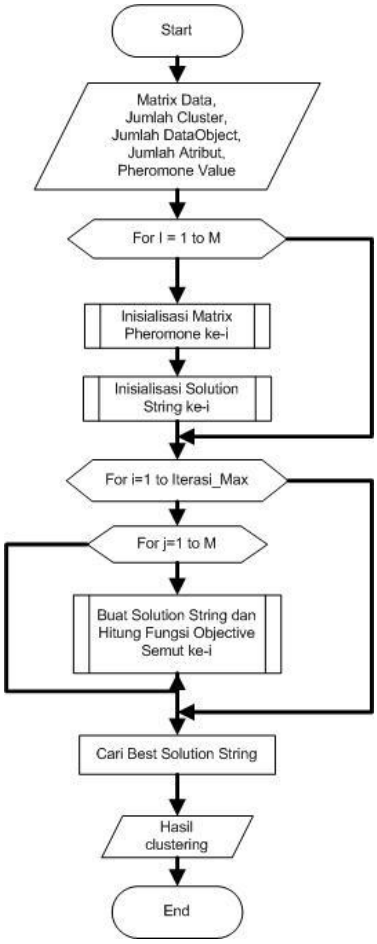

Gambar 6. Tampilan Clustering Dokumen

Gambar 6 dan Gambar 7 menunjukkan tampilan Clustering dokumen dengan fitur advanced search.




Gambar 7. Advanced Search dengan kata kunci "Fuzzy"

Gambar 8 menunjukkan langkah-langkah untuk melakukan klastering menggunakan Ant Colony Optimization.




Gambar 8. Bagan Alir Algoritma klastering menggunakan *Ant Colony Optimization*

Pengujian sistem yang dilakukan adalah pengujian clustering dokumen, dengan studi kasus menggunakan data dokumen. Pengujian ini dilakukan sehingga dapat membuktikan kinerja sistem aplikasi yang sudah disusun pada tahap implementasi.

Untuk pengujian clustering menggunakan algoritma *Ant Colony Optimization* menggunakan parameter sebagai berikut Jumlah cluster (K) = 5, Jumlah Iterasi = 5, $\alpha = 1$, $\beta = 2$, Jumlah Semut=5, $\rho = 0.1$, dan $q_0 = 0.01$.

Hasil pengujian clustering dokumen menggunakan algoritma Ant Colony Optimization yang didapatkan adalah waktu pemrosesan, nilai *variance* yang menentukan cluster ideal dan nilai *Sum Squared Error*. Adapun hasil clusteringnya dapat dilihat pada tabel 1. Hasil cluster yang ideal berdasarkan dari perolehan nilai variance yang minimum. Sedangkan untuk pengujian cluster menggunakan *Sum Squared Error* dari sejumlah pengujian dokumen didapatkan cluster yang dihasilkan kurang ideal.

Tabel 1. Pengujian Clustering *Ant Colony Optimization*

| No | Jumlah Dokumen | waktu | Variance | SSE |
|---|---|---|---|---|
| 1 | 50 | 1m:42s | 7.621419E-5 | 200,49818 |
| 2 | 100 | 5m:58s | 5.0215316E-5 | 322,69882 |
| 3 | 200 | 9m:16s | 1.4765678E-4 | 675,2462 |
| 4 | 500 | 22m:22s | 1.3339706E-4 | 1147,4847 |
| 5 | 750 | 34m:31s | 4.945447E-4 | 529,68787 |

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

## 5. Kesimpulan

Dari hasil penelitian dan pengujian yang dilakukan pada sistem aplikasi clustering dokumen menggunakan algoritma *Ant Colony Optimization* dapat ditarik kesimpulan sebagai berikut:

1. Sistem clustering menggunakan algoritma *Ant Colony Optimization* menghasilkan cluster yang tidak pasti.
2. Clustering menggunakan Algoritma *Ant Colony Optimization* memerlukan waktu pemrosesan yang sangat lama, hal ini dipengaruhi oleh banyaknya jumlah agen semut dan jumlah iterasi yang diberikan terhadapnya.
3. Hasil clustering menggunakan *Ant Colony Optimization* merupakan hasil cluster yang ideal berdasarkan dari perolehan nilai variance yang minimum. Sedangkan untuk pengujian cluster menggunakan *Sum Squared Error* dari sejumlah pengujian dokumen didapatkan cluster kurang ideal.

## REFERENSI


[1]. Dorigo, M., Bonabeau, E., dan Theraulaz, G., 1999, "*Swarm Intelligence From Natural to Artificial Systems"*, Oxford University Press, New York, USA.

[2]. Shelokar, V.K., Jayaraman, V.K., dan Kulkarni, B.D., 2004, "An Ant Colony Approach for Clustering", *Analytica Chimica Acta 509*, 187–195.

[3]. Kao, Y dan Cheng, K, 2006, "An ACO-Based Clustering Algorithm", *Ant Colony Optimization and Swarm Intelligence*, Volume 4150/2006, pp340-347, Springer Berlin / Heidelberg

[4]. Nadler, M dan Smith, E.P., 1993, "*Pattern recognition Engineering*", John Wiley & Sons.,Inc., USA.

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

[6]. Gose, E., Johnsonbaugh, R., dan Jost, S., 1996, *Pattern recognition and Image Analysis*, Prentice Hall, USA.