GA-Based Document Clustering

Cornelius David Herianto (2015730034)*

Jurusan Teknik Informatika, Fakultas Teknologi Informasi dan Sains,

Universitas Katolik Parahyangan, Bandung

E-mail: 7315034@student.unpar.ac.id

Latar Belakang Masalah

Pengelompokan (*clustering*) merupakan teknik yang digunakan untuk membagi kumpulan data (*dataset*) ke dalam kelompok objek serupa. Setiap kelompok (*cluster*) terdiri dari objek-objek yang serupa satu sama lain berdasarkan suatu ukuran dan tidak serupa dengan objek-objek lain diluar kelompoknya.

Clustering merupakan salah satu teknik pembelajaran tak terarah (*unsupervised learning*) karena pembagian kelompoknya tidak berdasarkan sesuatu yang telah diketahui sebelumnya, melainkan berdasarkan kesamaan tertentu menurut suatu ukuran tertentu[1].

Salah satu algoritma pengelompokan yang paling sering digunakan adalah *K-means* yang dilakukan dengan cara membagi data ke dalam K kelompok. Kelompok tersebut dibentuk dengan cara meminimalkan jarak antara titik pusat *cluster* (*centroid*) dengan setiap anggota *cluster* tersebut. *K-means* sudah terbukti efektif dalam melakukan pengelompokan dalam situasi apapun. Namun, cara tersebut tetap saja memiliki kekurangan yaitu dapat terjebak dalam *local optima* tergantung dengan pemilihan *centroid* awal.[2]

Masalah *local optima* dapat ditangani menggunakan *Genetic Algorithm* (GA) yang telah terbukti efektif dalam menyelesaikan masalah pencarian dan optimasi. GA merupakan teknik pencarian heuristik tingkat tinggi yang menirukan proses evolusi yang secara alami

terjadi[3] berdasarkan prinsip *survival of the fittest*. Algoritma ini dinamakan demikian karena menggunakan genetika sebagai model pemecahan masalahnya.[4]

Dalam GA, parameter dari *search space* dikodekan dalam bentuk deretan objek yang disebut kromosom. Kumpulan kromosom tersebut lalu dikenal sebagai populasi. Pada awalnya, populasi dibangkitkan secara acak. Kemudian, akan dipilih beberapa kromosom menggunakan teknik *roulette wheel selection* berdasarkan fungsi *fitness*. Operasi dasar yang terinspirasi dari Ilmu Biologi seperti persilangan (*crossover*) dan mutasi (*mutation*) digunakan untuk membangkitkan generasi berikutnya. Proses seleksi, persilangan, dan mutasi ini berlangsung dalam jumlah generasi tertentu atau sampai kondisi akhir tercapai.

Fungsi *fitness* tidak hanya berfungsi untuk menentukan seberapa baik solusi yang dihasilkan namun juga menentukan seberapa dekat solusi tersebut dengan hasil yang optimal.[4] Oleh karena itu, diperlukan fungsi *fitness* yang cocok sehingga GA dapat menghasilkan keluaran yang optimal. Pada masalah *clustering* menggunakan GA, maka fungsi *fitness* yang digunakan harus bisa menggambarkan bahwa seluruh elemen sudah berada dalam *cluster* yang terbaik dan sudah sesuai.

Dasar Teori

Pengelompokan dokumen berkaitan erat dengan dua bidang ilmu dalam informatika. Pengelompokan dalam informatika merupakan bagian dari bidang pembelajaran mesin. Dalam pembelajaran mesin, terdapat dua jenis pengelompokan, yaitu *clustering* dan *classification*. *Clustering* merupakan salah satu jenis pembelajaran tak terarah (*unsupervised learning*) karena setiap elemen dikelompokkan berdasarkan karakteristik dari elemen tersebut. Sedangkan *classification* merupakan jenis pembelajaran terarah (*supervised learning*) karena setiap elemen dikelompokan berdasarkan label yang telah ditentukan sebelumnya. Pada penelitian ini, jenis pengelompokan yang akan digunakan adalah *clustering*.

Domain masalah dalam penelitian ini adalah dokumen sehingga sangat erat kaitannya

dengan bidang temu kembali informasi (*information retrieval*). Sebelum dapat diolah lebih lanjut, dokumen-dokumen yang telah ada harus diolah sehingga menjadi suatu bentuk dengan nilai informasi yang dapat dimanipulasi oleh komputer. Biasanya dokumen akan direpresentasikan ke dalam bentuk vektor yang disebut dengan model ruang vektor.

Temu Kembali Informasi

Temu kembali informasi adalah bidang ilmu yang berurusan dengan representasi, penyimpanan, pengolahan, dan akses terhadap informasi.[5] Pada penelitian ini, temu kembali informasi memiliki peran untuk mengubah dokumen yang semula berbentuk teks menjadi sesuatu yang memiliki nilai informasi agar nantinya bisa diproses lebih lanjut (dalam kasus ini dengan pengelompokan dokumen).

Dalam penelitian ini, temu kembali informasi digunakan untuk merepresentasikan dokumen ke dalam model ruang vektor agar bisa dilakukan pengelompokan. Tahap pertama yang dilakukan adalah membentuk kosa kata (*vocabulary*) yang berisi seluruh istilah berbeda yang ada di setiap dokumen yang akan diindeks. Kemudian, untuk setiap dokumen akan dibentuk suatu indeks yang terdiri dari pasangan antara istilah di dokumen tersebut dan jumlah kemunculannya.

Selanjutnya, setiap jumlah kemunculan suatu istilah dalam sebuah dokumen akan diubah menjadi suatu bobot tertentu berdasarkan teknik pemodelannya. Dalam penelitian ini, digunakan 2 teknik pembobotan yaitu bobot frekuensi dan bobot tf-idf.

Bobot frekuensi

Bobot frekuensi merupakan teknik pembobotan yang sangat sederhana karena bobotnya merupakan jumlah kemunculan istilah tersebut dalam dokumen. Bobot frekuensi dapat digambarkan dalam rumus:

$$w_i = tf_i \tag{1}$$

dengan w_i merupakan bobot istilah ke-i dan tf_i merupakan frekuensi kemunculan istilah ke-i pada dokumen.

Bobot TF-IDF

Pada bobot frekuensi, bobot hanya dihitung berdasarkan kemunculan istilah dalam dokumen itu sendiri. Namun dalam bobot TF-IDF (*term frequency-inverse document frequency*), bobot juga dihitung berdasarkan kemunculan istilah pada himpunan dokumen. Metode ini sangat populer digunakan oleh sistem rekomendasi berbasis teks.[6] Rumus dari TF-IDF adalah sebagai berikut:

$$w_i = tf_i \times log\frac{N}{N_i} \tag{2}$$

dengan w_i merupakan bobot istilah ke-i, tf_i merupakan frekuensi kemunculan istilah ke-i pada dokumen, N menyatakan jumlah anggota himpunan dokumen, dan N_i menunjukkan frekuensi dokumen dari istilah ke-i (jumlah dokumen pada himpunan dokumen yang memuat istilah ke-i).

Berdasarkan rumus tersebut, maka dapat ditarik dua kesimpulan yaitu:

- Semakin sering suatu istilah muncul di suatu dokumen, maka semakin representatif istilah tersebut terhadap isi dokumen.
- Semakin banyak dokumen yang memuat suatu istilah, maka nilai informasi istilah tersebut semakin kecil.

Metode penetapan bobot TF-IDF dianggap sebagai metode yang berkinerja baik karena mempertimbangkan frekuensi kemunculan istilah baik secara lokal (TF) maupun global (IDF).

Model ruang vektor

Model ruang vektor adalah representasi dari koleksi dokumen sebagai vektor dalam ruang vektor yang umum.[7] Model ruang vektor ini biasanya digunakan dalam sejumlah opera-

si pencarian informasi mulai dari penilaian dokumen pada *query*, klasifikasi dokumen dan pengelompokan dokumen.

Dalam pengolahan model ruang vektor, dibutuhkan cara untuk menghitung kesamaan antara dua ruang vektor. Dalam pengelompokan dokumen, apabila kesamaan antara dua buah vektor semakin besar, maka peluang kedua vektor tersebut berada dalam sebuah kelompok yang sama akan semakin besar. Dalam penelitian ini, digunakan dua cara untuk menghitung kesamaan antara dua ruang vektor yaitu menggunakan Jarak Euclidean (*Euclidean distance*) dan persamaan cosinus (*cosine similarity*).

Jarak Euclidean

Jarak Euclidean atau biasa disebut jarak garis lurus merupakan metode paling banyak digunakan untuk menghitung jarak antara dua buah vektor. Secara umum, rumus dari Jarak Euclidean adalah sebagai berikut:

$$d_{ij} = \sqrt{\sum_{v=1}^{N}(x_{vi} - x_{vj})^2} \tag{3}$$

dengan d_{ij} adalah jarak antara vektor ke-i dengan vektor ke-j, N adalah jumlah dimensi pada vektor, dan x_{vi} adalah nilai dimensi ke-v dari vektor ke-i.

Persamaan Cosinus

Persamaan cosinus merupakan normalisasi hasil kali titik dengan panjang masing-masing vektor. Persamaan cosinus memiliki persamaan sebagai berikut:

$$s_{ij} = \frac{i \cdot j}{\parallel i \parallel \times \parallel j \parallel} \tag{4}$$

dengan s_{ij} adalah kesamaan antara vektor ke-i dengan vektor ke-j, i adalah vektor ke-i, dan j adalah vektor ke-j. Persamaan ini menjelaskan bahwa semakin kecil sudut antara dua

vektor, maka tingkat kemiripannya semakin besar.

Pembelajaran Mesin

Pembelajaran mesin merupakan sistem yang dapat beradaptasi dengan keadaan baru dan dapat mengenali pola[8]. Sistem yang telah belajar akan meningkatkan kinerjanya pada tugas-tugas di masa yang akan datang setelah melakukan pengamatan tentang lingkungannya.

Pengelompokan

Tugas pengelompokan (*clustering*) merupakan salah satu bagian dari pembelajaran mesin. Pengelompokan terdiri dari dua jenis berdasarkan metode pembelajarannya. *Classification* merupakan jenis pembelajaran terarah karena sudah diberikan label sejak awal, lalu dilakukan pengelompokan berdasarkan label untuk setiap kelompoknya. Sedangkan *clustering* merupakan jenis pembelajaran tak terarah karena tidak diberikan label sejak awal sehingga pengelompokan dilakukan berdasarkan suatu kesamaan tertentu.

K-Means

K-means merupakan algoritma pengelompokan yang paling populer digunakan saat ini. algoritma ini membagi data ke dalam K *cluster*. Setiap *cluster* direpresentasikan dengan titik tengahnya (*centroid*). Setiap iterasi, titik tengah akan dihitung sebagai rata-rata dari semua titik data dari *cluster* tersebut. Rumus untuk menghitung *centroid* adalah sebagai berikut:

$$\mu_i = \frac{1}{N_i} \sum_{q=1}^{N_i} x_q \tag{5}$$

dengan μ_i merupakan *centroid* ke-i, N_i merupakan jumlah titik data pada *cluster* ke-i, dan x_q merupakan titik ke-q pada *cluster* ke-i.

Algoritma K-means diawali dengan penentuan *centroid* secara acak. Pada setiap iterasi, setiap data ditetapkan pada *cluster* yang memiliki *centroid* dengan jarak terdekat dari ti-

Algorithm 1 K-Means

Input: S (himpunan titik data), K (Jumlah *cluster*)
Output: himpunan *cluster*

1: Pilih K titik data sebagai himpunan awal *centroid*.
2: **repeat**
3: Bentuk K *cluster* dengan menempatkan setiap titik data ke *cluster* dengan *centroid* terdekat.
4: Hitung ulang *centroid* untuk setiap *cluster*.
5: **until** *Centroid* tidak berubah.

tik data tersebut, kemudian posisi *centroid* dari setiap *cluster* akan dihitung ulang dengan Persamaan 5. Iterasi akan terus diulang sampai posisi dari semua *cluster* tidak berubah.

Algoritma Genetika

Algoritma genetika (GA) adalah suatu algoritma pencarian yang terinspirasi dari proses seleksi alam yang terjadi secara alami dalam proses evolusi. GA merupakan varian dari *stochastic beam search* di mana sifat dari sebuah keturunan dihasilkan dengan menggabungkan sifat dua induk dan bukan hanya dengan memodifikasi sifat satu induk.[8]

Ada beberapa istilah yang digunakan dalam algoritma genetika diantaranya kromosom, seleksi, persilangan, mutasi, dan fungsi *fitness*.

Kromosom

Dalam GA, kromosom adalah himpunan parameter yang mendefinisikan suatu solusi yang diusulkan. Kromosom biasanya direpresentasikan sebagai string yang berisi kumpulan nilai (gen), meskipun berbagai struktur data lainnya juga digunakan.

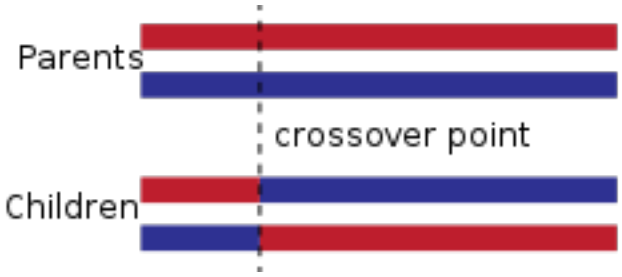
Seleksi

Seleksi dalam algoritma genetika bertugas untuk memilih kromosom dari populasi untuk proses persilangan. Salah satu teknik yang populer digunakan dalam seleksi adalah *roulette-*

wheel selection. Teknik ini memungkinkan terjadinya pemilihan berdasarkan fungsi *fitness* sehingga kromosom yang memiliki fungsi *fitness* lebih besar memiliki peluang lebih besar untuk terpilih dan menjadi induk dari generasi selanjutnya.

Persilangan

Persilangan adalah operasi genetik yang digunakan untuk menggabungkan informasi genetik dari dua induk untuk menghasilkan keturunan baru. Teknik persilangan yang digunakan dalam penelitian ini adalah *Single-point crossover*. Dalam teknik ini, sebuah titik pada kedua induk dipilih untuk menjadi titik persilangan (*crossover point*). Bit yang berada di sebelah kanan titik persilangan bertukar antara kedua kromosom induk seperti yang ditunjukkan pada Gambar 1.



Gambar 1: *Single-point crossover*

Mutasi

Mutasi adalah suatu operator genetik yang digunakan untuk mempertahankan keragaman genetik dari satu generasi populasi dalam algoritma genetika. Mutasi mengubah satu atau beberapa nilai dalam gen. Mutasi terjadi berdasarkan probabilitas mutasi yang sudah ditentukan sebelumnya. Probabilitas ini seharusnya bernilai kecil, karena jika terlalu besar maka akan menjadi sama dengan algoritma pencarian acak primitif (*primitive random search*). Pada penelitian ini, kromosom memiliki gen yang bernilai non-biner. Oleh karena itu, mutasi dilakukan dengan memilih sebuah bilangan acak antara batas bawah dan batas atas nilai suatu gen.

Fungsi *Fitness*

Fungsi *fitness* adalah fungsi objektif yang digunakan untuk mengukur seberapa baik suatu kromosom sebagai calon solusi. Sebuah fungsi *fitness* harus bisa memperkirakan seberapa dekat sebuah calon solusi dengan solusi yang optimal. Dalam algoritma genetika, fungsi *fitness* juga digunakan dalam proses seleksi (*roulette-wheel selection*) untuk menentukan seberapa baik suatu kromosom untuk menjadi induk dari generasi berikutnya.

Algorithm 2 Algoritma Genetika

function Algoritma-Genetika(*populasi*,Fung-Fitness) **returns** solusi berupa individu

inputs: *populasi*, himpunan individu
Fung-Fitness, fungsi yang mengukur *fitness* dari tiap individu

```
1: repeat
2:    populasi_baru ← himpunan kosong
3:    for i=1 to Size(populasi) do
4:       x ← Seleksi-acak(populasi, Fung-Fitness)
5:       y ← Seleksi-acak(populasi, Fung-Fitness)
6:       anak ← Persilangan(x, y)
7:       if probabilitas mutasi then
8:          Mutasi(anak)
9:       end if
10:      tambahkan anak ke populasi_baru
11:   end for
12:   populasi ← populasi_baru
13: until kriteria berakhir dipenuhi atau sampai jumlah generasi tertentu
14: return individu terbaik dalam populasi, berdasarkan Fung-Fitness
```

function Persilangan(x,y) **returns** anak berupa individu

inputs: x, y, individu induk

```
1: n ← Length(x)
2: c ← angka acak antara 1 sampai n
3: return Append(Substring(x,1,c), Substring(y,c+1,n))
```

Proses pencarian dalam algoritma genetika

Seperti yang disebutkan dalam Algoritma 2, algoritma genetika dimulai dengan menginisialisasi suatu populasi (biasanya dibangkitkan secara acak jika tidak diketahui heuristik masalahnya). Lalu, akan dibangkitkan populasi baru untuk generasi berikutnya dengan cara mengambil dua kromosom secara acak dengan teknik *roulette-wheel selection*. Setelah itu akan dilakukan persilangan dengan teknik *single-point crossover*. Teknik ini dilakukan dengan cara menentukan sebuah titik potong c yang diambil secara acak antara angka 1 sampai panjang kromosom n. Keturunan dari kedua induk tersebut akan memiliki kromosom induk pertama dari gen ke-1 sampai gen ke-c dan dari induk kedua mulai dari gen ke-(c+1) sampai gen ke-n. Setelah itu, untuk apabila terjadi mutasi, maka salah satu gen dari anak akan diubah nilainya. Iterasi ini akan dilakukan terus-menerus hingga kriteria berhenti tercapai atau sudah mencapai batas jumlah generasi tertentu.

GA dalam Pengelompokan

Meskipun pada umumnya algoritma *K-means* digunakan dalam pengelompokan, tetapi ternyata algoritma *K-means* masih memiliki kekurangan yaitu masih dapat terjebak pada *local optimum*. Oleh karena itu, pada penelitian ini digunakan algoritma genetika sebagai solusi dari permasalahan *local optimum*. Algoritma genetika memiliki kemampuan untuk menghindari terjadinya *local optimum* karena algoritma genetika merupakan algoritma pencarian yang bersifat stokastik karena memperbolehkan terjadinya variasi acak dalam proses pencarian solusinya. Oleh karena itu, diharapkan pengelompokan berbasis GA dapat menghasilkan solusi yang lebih baik dibandingkan dengan pengelompokan pada umumnya yang menggunakan algoritma *K-means*.

Perancangan

Perangkat lunak yang akan dibangun dalam penelitian ini merupakan perangkat lunak untuk pengelompokan berbasis algoritma genetika (*GA-based document clustering*). Oleh karena itu, ada beberapa hal yang perlu dirancang untuk mengadaptasi GA agar bisa melakukan fungsi pengelompokan diantaranya: representasi kromosom, fungsi *fitness*, dan operasi genetika.

Representasi Kromosom

Setiap *string* kromosom merupakan deretan bilangan riil yang merepresentasikan K titik pusat *cluster* (*centroid*). Dalam ruang N dimensi, panjang dari kromosom akan menjadi $N \times K$ kata. N kata pertama merepresentasikan N dimensi dari *centroid* pertama, N kata selanjutnya merepresentasikan N dimensi dari *centroid* kedua, dan seterusnya.

Fungsi *Fitness*

Perhitungan *fitness* dalam penelitian ini terdiri dari dua tahap. Pada tahap pertama, terjadi pembentukan *cluster* berdasarkan titik pusat yang terkandung dalam kromosom. Hal ini dilakukan dengan menetapkan setiap titik $x_i, i = 1, 2, ..., n$ ke dalam sebuah *cluster* C_j dengan *centroid* z_j sehingga

$$\| x_i - z_j \| < \| x_i - z_p \|, p = 1, 2, ..., K, \text{ and } p \neq j. \tag{6}$$

Setelah proses pengelompokan selesai, titik pusat yang terkandung dalam kromosom diganti dengan rata-rata titik dari tiap *cluster*. Dengan kata lain, untuk *cluster* C_i, *centroid* baru z_i^* dapat dihitung dengan rumus:

$$z_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j, i = 1, 2, ..., K. \tag{7}$$

dengan z_i^* merupakan titik pusat *cluster* ke-i, n_i merupakan jumlah anggota *cluster* ke-i, dan x_j merupakan titik ke-j yang merupakan anggota dari *cluster* ke-i. z_i^* ini akan menggantikan z_i sebelumnya di kromosom. Selanjutnya akan dihitung sebuah *clustering metric* M dengan rumus:

$$M = \sum_{i=1}^{K} M_i,$$

$$M_i = \sum_{x_j \in C_i} \parallel x_j - z_i \parallel$$

Lalu, fungsi *fitness* akan didefinisikan sebagai $f = 1/M$, sehingga maksimalisasi terhadap nilai f akan meminimalkan nilai M.

Operasi Genetik

Ada beberapa operasi genetik yang akan dibahas, diantaranya: inisialisasi populasi, seleksi, persilangan, dan mutasi.

Inisialisasi Populasi

K *centroid* yang terkandung dalam kromosom pada mulanya dipilih secara acak sebanyak K titik dari keseluruhan himpunan data. Lalu proses ini diulang sebanyak P kali di mana P merupakan ukuran populasi yang diinginkan.

Seleksi

Proses seleksi ini terjadi berdasarkan konsep *survival of the fittest* yang diadaptasi dari sistem genetika alami. Dalam penelitian ini, calon induk dari generasi selanjutnya dipilih dengan menggunakan teknik *roulette-wheel selection* yang merupakan salah satu teknik yang digunakan karena mengimplementasikan *proportional selection strategy.*

Persilangan

Persilangan dalam penelitian ini terjadi terhadap dua induk dengan satu titik potong. Misalkan kromosom memiliki panjang l, sebuah angka acak akan diambil sebagai titik potong dalam batas [1,$l-1$]. Bagian kromosom sebelah kanan titik potong akan ditukar antara kedua induk sehingga menghasilkan dua individu keturunan.

Mutasi

Setiap kromosom mengalami mutasi dengan probabilitas mutasi tetap μ_c. Setelah itu, ditentukan gen mana yang akan mengalami mutasi dengan mengambilnya secara acak. Perubahan nilai gen tersebut juga merupakan angka acak yang diambil mulai dari angka 0 sampai dengan jumlah kemunculan kata keseluruhan dalam *vocabulary*.

Pustaka

(1) Raposo, C.; Antunes, C. H.; Barreto, J. P. Automatic Clustering using a Genetic Algorithm with New Solution Encoding and Operators. International Conference on Computational Science and Its Applications. 2014; pp 92–103.

(2) Maulik, U.; Bandyopadhyay, S. Genetic algorithm-based clustering technique. *Pattern recognition* **2000**, *33*, 1455–1465.

(3) Holland, J. H. Genetic algorithms. *Scientific american* **1992**, *267*, 66–73.

(4) Sivanandam, S.; Deepa, S. *Introduction to Genetic Algorithms*; Springer Science & Business Media, 2007.

(5) Baeza-Yates, R.; Ribeiro-Neto, B., et al. *Modern information retrieval*; ACM press New York, 1999; Vol. 463.

(6) Aizawa, A. An information-theoretic perspective of tf–idf measures. *Information Processing & Management* **2003**, *39*, 45–65.

(7) Schütze, H.; Manning, C. D.; Raghavan, P. *Introduction to information retrieval*; Cambridge University Press, 2008; Vol. 39.

(8) Russell, S. J.; Norvig, P. *Artificial intelligence: a modern approach*; Malaysia; Pearson Education Limited,, 2016.