# BAB II

# TINJAUAN PUSTAKA

## A. Penelitian Terdahulu

Penelitian yang telah dilakukan dan menjadi acuan dalam penelitian ini dapat dilihat dalam Tabel 2.1.

Tabel 2.1 Penelitian terdahulu.

| No | Penulis | Judul | Metode | Hasil |
|---|---|---|---|---|
| 1. | (Fitriani, Musdholifah and Hartati, 2018) | *Adaptive Unified Differential Evolution for Clustering* | *Adaptive Unified Differential Evolution* (AuDE) | Pada penelitian ini *clustering* menggunakan *Adaptive Unified Differential Evolution* (AuDE). Metode clustering AuDE diuji menggunakan 4 dataset. Dataset yang diuji yaitu Iris, Wine, Glass dan Ecoli. *Silhouette Index* dan CS Measure merupakan fungsi fitness yang digunakan sebagai alat ukur kualitas hasil clustering. |
| 2. | Ramdani and Firmansyah, (2018) | *Clustering Application for UKT Determination Using Pillar K-Means Clustering Algorithm and Flask Web Framework* | Algoritma *K-Means* dan Flask Web Framework | Pada penelitian ini evaluasi *clustering* menggunakan *koefisien silhouette.* Nilai *koefisien silhouette cluster* tertinggi yang akan diambil untuk setiap nilai α, β dan *k*. Sehingga menghasilkan nilai *k*=3 adalah solusi *cluster* terbaik pada proses *clustering* data. |

Tabel 2.2 (Lanjutan).

| No | Penulis | Judul | Metode | Hasil |
|---|---|---|---|---|
| 3. | (Setiawan, Herwindiati and Sutrisno, 2019) | Algoritma Genetika dengan *Roulette Wheel Selection* dan *Arithmetic Crossover* untuk Pengelompokan. | Algoritma genetika dengan *Roulette Wheel Selection* dan *Arithmetic Crossover* | Menganalisis Algoritma Genetika (AG) dengan *roulette wheel selection* dan *arithmetic crossove*r untuk pengelompokan data. Data diujikan yaitu citra bunga dengan beberapa skenario. Penelitian ini dalam pengelompokan gambar bunga dengan warna yang berbeda memiliki hasil yang baik, sedangkan pengelompokan gambar bunga dengan warna yang sama tidak memberikan hasil yang baik. Percobaan dilakukan pada setiap skenar io untuk mengetahui pengaruh parameter yang digunakan terhadap nilai *fitness* yang diperoleh, hasilnya adalah *clustering* dengan parameter karakteristik warna, parameter dengan nilai *fitness* terbesar adalah jumlah populasi = 100, iterasi = 200, dan mutasi = 0,02. Sedangkan *clustering* dengan karakteristik warna tambah tekstur, parameter dengan nilai *fitness* terbesar adalah jumlah populasi=200, *iterasi*=300, dan mutasi=0.02. Sehingga AG menghasilkan kinerja yang baik dalam pengelompokan data, dengan memperhatikan parameter yang digunakan untuk memperoleh hasil yang maksimal. |

Tabel 2.3 (Lanjutan)

| No | Penulis | Judul | Metode | Hasil |
|---|---|---|---|---|
| 4. | Kuntjoro, Setiawan and Perdana, (2018) | Algoritma Genetika Untuk Optimasi *K-Means* Clustering Dalam Pengelompokan Data Tsunami | Algoritma Genetika, *K-Means* | Menggunakan *silhouette coefficient* untuk mengevaluasi hasil *clustering* menggunakan K-*means* dan K-*means* optimasi Algoritma Genetika. Nilai validasi *silhouette coefficient* Algoritma Genetika dan K-*means* yaitu 0,9959 sedangkan untuk K-*means* nilainya yaitu 0,8831. Hasil optimal pada metode Algoritma Genetika dan K-*means* dicapai pada parameter jumlah *popsize* 50, generasi 70 dan gabungan Cr=0,9 Mr=0,1. |
| 5. | Akay, Tekeli and Yüksel, (2020) | *Genetic Algorithm with New Fitness Function for Clustering* | Algoritma genetika dengan *new fitness* | Menganalisis AG dengan *new fitness* untuk *clustering. New fitness* didefinisikan dengan menambahkan fungsi *silhouette* yang menunjukan data berada di *cluster* yang benar. Meminimalkan rasio jarak dalam *cluster* ke jarak antar *cluster*. Data uji menggunakan *dataset*. Hasil analisis menunjukkan algoritma ini dapat menghasilkan *clustering* yang lebih baik dibandingkan beberapa algoritma *clustering* lainnya. Oleh karena itu, penggunaan fungsi *new fitness* memastikan konvergensi ke optimum global. |

## B. Kajian Teori

1. *Clustering* (Pengelompokan)

*Clustering* adalah teknik statistik banyak variasi yang dimulai dengan kumpulan data term asuk informasi tentang beberapa upaya dan variabel untuk membagi status data kedalam kelompok yang relatif sama.

*Clustering* sebagai salah satu teknik pengenalan pola yang popular dan telah banyak digunakan dalam berbagai bidang, seperti *web mining*, segmentasi citra, *machine learning*, pengenalan biometrik, teknik elektro, teknik mesin, penginderaan jauh, dan genetika (Gu dan Lu, 2012). *Clustering* adalah tema penelitian yang paling penting dalam ruang lingkup *data mining* dan sangat berpengaruh untuk banyak aplikasi, seperti pemasaran, biologi, teknik industry, kedokteran, dan *image processing* (Yang dan Chi, 2005).

2. Evaluasi

a. *Silhouette Coefficient* (Koefisien Silhouette)

*Silhouette Coefficient* adalah salah satu metode evaluasi yang digunakan untuk melihat seberapa optimal data dalam suatu *cluster*. Evaluasi *clustering* berfungsi untuk mengetahui seberapa tepat suatu data dikelompokan. Nilai hasil perhitungan *silhouette coefficient* memiliki rentang dari -1 hingga 1. Jika s*ilhouette coefficient* bernilai 1 artinya *cluster* yang dihasilkan tepat dan objek ke-*i* telah berada dalam *cluster* yang tepat. Tetapi jika nilai s*ilhouette coefficient* adalah 0, maka objek ke-*i* berada antara dua *cluster* sehingga tidak jelas harus diletakkan kedalam *cluster* yang mana. Jika nilai s*ilhouette coefficient* adalah -1 berarti *cluster* yang telah dihasilkan tidak baik, sehingga objek ke-*i* lebih tepat diletakan ke dalam *cluster* yang lain (Dani, Wahyuningsih and Rizki, 2019).

Perhitungan nilai *silhouette index* dari sebuah data ke-*i* dibutuhkan untuk menghitung nilai *silhouette coefficient.* Nilai

*silhouette coefficient* diperoleh dengan mencari nilai maksimal dari nilai *Silhouette Index Global* dari jumlah *cluster* sampai dengan jumlah *cluster n-1*. Metode ini merupakan gabungan dari separasi dan kohesi (Dewi and Pramita, 2019). Rumus untuk menghitung nilai Silhouette Index dari suatu objek *i* yaitu sebagai berikut:

$$s(i) = \frac{b(i)-a(i)}{max\{(a),(b)\}} \quad \text{.....(2.14)}$$

$$a_i = \frac{\sum d_{i,j}}{nC_i}, i, j \in C_i \quad \text{.....(2.15)}$$

$$b(i) = min_{C_k \neq C_i} \left\{ \frac{\sum d_{i,k}}{nc_k}, i, C_i \; and \; k \in C_k \right\} \quad \text{.....(2.16)}$$

Keterangan:

$S_i$ = Indeks Silhouette dari data i (*silhouette width)*

$a_i$ = rata-rata silhouette (jarak data ke-i dengan data lain dalam *cluster* yang sama)

$b_i$ = jarak rata-rata antara data ke-i dengan data yang berada dalam *cluster* berbeda.

$d_{i,j}$ = jarak antara data ke-i dan data ke-j, $nc_i$ dan $nc_k$ = jumlah data *cluster* ke-i dank e-k.

Setelah lebar silhouette *S*i setiap data diperoleh, maka akan dihitung nilai *index silhouette* dihitung dengan menggunakan persamaan (2.17). Range nilai SC yaitu [-1, 1], dimana 1 menunjukan hasil *clustering* bagus dan -1 buruk.

$$SC = \frac{1}{n}\sum S_i \quad \text{.....(2.17)}$$

Keterangan:

n = jumlah data

3. *Euclidean Distance*

*Euclidean distance* merupakan salah satu metode yang dipakai untuk mengukur jarak dari satu titik ke titik yang lain yang berbeda (Agusta, 2018). Metode *Euclidean Distance* bekerja secara efektif dalam mengelompokan data dengan menghitung kemiripan data (Aditya *et al.*,

2021). Sehingga pada penelitian yang akan dilakukan untuk mengukur jarak dari setiap objek ke masing-masing *centroid* dari setiap *cluster* menggunakan rumus *Euclidean distance* sebagai berikut:

$$d(x, y) = \sqrt{\sum_{k=1}^{n}(X_k - Y_k)^2} \quad \text{..... (2.1)}$$

Keterangan:

Xk = nilai x pada atribut ke-k

Yk = nilai y pada atribut ke-k

Keanggotaan *cluster* ditentukan dengan cara melihat jarak data ketitik *cluster* yang terpendek, sehingga data yang memiliki jarak terpendek dengan pusat *cluster* dialokasikan ke dalam *cluster* tersebut.

Pusat *cluster* pada penelitian yang akan dilakukan ditentukan dengan cara membangkitkan secara random nilai aktivasi *threshold* dengan rentang nilai (0-1).

4. Algoritma Genetika (AG)

Algoritma Genetika merupakan suatu algoritma yang termotivasi dari suatu teori evolusi yang diciptakan oleh Charles Darwin sebagai sarana untuk menyelesaikan permasalahan yang rumit (komplek). AG menerapkan seleksi alami dengan penjabaran dalam bentuk step-step prosedur kromosom buatan yang dimulai dari satu populasi ke populasi baru.(Istianto *et al.*, 2021).

Algoritma Genetika mengkombinasikan berbagai pilihan solusi terbaik secara *random* dalam suatu kelompok untuk memperoleh generasi solusi terbaik. AG tidak mudah terperangkap dalam optimasi local karena memiliki sifat pencarian global. Optimasi local menghasilkan pusat *cluster* dengan sifat seperti perulangan sebelumnya sehingga hasil pengelompokan kurang baik dan perlu dihindari dan hasil *clustering* semakin tidak baik jika dalam data observasi masih terdapat *Outlier*.(Setiawan, Herwindiati and Sutrisno, 2019).

Menurut (Setiawan, Herwindiati and Sutrisno, 2019) adapun tahap-tahap dalam AG yaitu sebagai berikut:

a. Inisialisasi Populasi (Populasi Awal)

Populasi awal merupakan langkah awal yang harus dilakukan dalam AG. Populasi awal akan dibangkitkan secara random sehingga menghasilkan solusi awal (Rindengan, Cholissodin and Adikara, 2018).

Ada beberapa komponen yang terdapat didalam populasi yaitu gen yang berada di dalam kromosom dan kumpulan kromosom disebut dengan individu. Pada penelitian ini setiap individu hanya terdiri dari satu anggota. Dalam AG terdapat beberapa jenis kromosom diantaranya kromosom *string,* kromosom *float,* dan kromosom biner. Pada penelitian yang akan dilakukan menggunakan jenis kromosom *float.* Kromosom *float* terdiri dari gen-gen dengan pecahan. Pada jenis kromosom *integer* juga bisa digolongkan.

Struktur kromosom terdiri dari *centroid* dari tiap *cluster*. Populasi awal terdiri dari kromosom yang dipilih secara random dari ruang solusi. Oleh karena itu, n*k kromosom yang terpilih pada populasi awal n adalah ukuran populasi pada setiap generasi (Akay, Tekeli and Yüksel, 2020).

Populasi AG juga merupakan string yang terdiri dari titik-titik data yang dipilih secara random. Awal titik-titik data berperan sebagai pusat *cluster*. Jumlah kromosom dalam suatu populasi seimbang dengan jumlah titik-titik data dalam kumpulan data (Geetha Lekshmy V, Anusree P K, 2018). Kromosom berisi informasi solusi dari demikian banyak kemungkinan solusi masalah yang dihadapi (Ramadhania and Rani, 2021).

b. Fungsi *Fitness*

Menurut (Setiawan, Herwindiati and Sutrisno, 2019) Nilai *fitness* adalah suatu proses untuk menilai setiap populasi dengan menghitung nilai *fitness* setiap kromosom. Kinerja suatu individu dievaluasi berdasarkan fungsi objektif. Fungsi objektif yang digunakan

adalah *Minimum distance*, fungsi objektivitas *Minimum Distance* digunakan untuk mengecek validasi setiap pusat *cluster* yang dihasilkan setiap kromosom. Semakin kecil nilai objektifitas maka semakin besar nilai fitness yang dihasilkan, dengan besarnya nilai *fitness* maka kemungkinan kromosom terpilih pada tahap seleksi semakin besar.

Pada penelitian ini menggunakan *new fitness*. Adapun rumus untuk mencari nilai *fitness* menurut (Akay, Tekeli and Yüksel, 2020) adalah sebagai berikut:

1) Persamaan yang digunakan untuk menghitung nilai *fitness* menggunakan rumus *new fitness* sebagai berikut:

$$FF = \frac{Sum\ (BC)}{Sum\ (WC)} + SW \quad \text{.....(2.2)}$$

Keterangan:

$BC$ = Jarak antar *cluster*

$WC$ = Jarak dalam *cluster*

$SW$ = Lebar *Silhouette*

BC jarak antara *m*th dan *n*th cluster n (m,n= 1, 2, …,k) dihitung dengan rumus:

$$BC_{m,n} = \sqrt{\sum_{i=1}^{q}\sum_{j=1}^{r}\frac{(X_i - X_j)^2}{(q*nr)}} \text{.....(2.3)}$$

Keterangan:

$q, r$ = Jumlah masing-masing elemen dalam *cluster* $m, n$ $(m = n)$

Jarak BC antara m, n dan n, m adalah sama dan jarak BC adalah 0.

Jarak WC dari mth *cluster* (m= 1, 2, …, k) dihitung dengan rumus:

$$WC_{m,n} = \sqrt{\sum_{i=1}^{q}\sum_{j=1}^{q}\frac{(X_i - X_j)^2}{(q*q)}} \text{.....(2.4)}$$

Keterangan:

$q =$ Jumlah elemen dalam *cluster* m

Jarak total $BC$ dan $WC$ didefinisikan dengan rumus sebagai berikut:

$$Sum(BC) = \sum_{m=1}^{k-1}\sum_{n=m+1}^{k} BC_{m,n} \text{.....(2.5)}$$

$$Sum(WC) = \sum_{m=1}^{k} WC_m \text{.....(2.6)}$$

Rumus yang digunakan untuk menghitung nilai rata-rata *silhouette* adalah sebagai berikut:

$$SW_i = \frac{b_i - a_i}{\max(b_i, a_i)} \text{.....(2.7)}$$

Jika nilai mendekati 1 pengamatan terkelompok dengan baik, jika mendekati -1 pengamatan buruk berkerumunan.

$a_i =$ Jarak rata-rata pengamatan i dan semua pengamatan lain dalam *cluster* yang sama. $a_i$ dihitung dengan rumus:

$$a_i = \frac{1}{s(C(i))} \sum_{j \in C(i)} dist(i,j) \text{.....(2.8)}$$

$b_i =$ Jarak rata-rata antara pengamatan i dan pengamatan dalam *cluster* terdekat

$$b_i = \min_{C_k \in C \setminus C(i)} \sum_{j \in (C_k)} \frac{dist(i,j)}{s(C_k)} \text{.....(2.9)}$$

Keterangan:

$C(i) =$ *cluster* yang memiliki observasi i

Dist (i, j) = Jarak pengamatan i dan j

$s(C) =$ Jumlah observasi pada *cluster* C

SW dihitung dengan rumus:

$$SW = \frac{\sum_{i=1}^{s} SW_i}{s} \text{.....(2.10)}$$

Keterangan:

s = ukuran sampel

$SW$ terletak pada interval [-1, 1]

*c. Selection*

Proses seleksi berguna agar dapat memilih kualitas kromosom yang baik sehingga dapat meneruskan proses *crossover*. Proses seleksi berfungsi untuk menyeleksi kromosom dengan kualitas yang baik sehingga dapat dilanjutkan ke proses *crossover*. Ada beberapa macam teknik seleksi yaitu *Roulette Wheel Selection, Elitism Selection, Rank*

*Based Selection, dan Steady State Selection* (Ramadhania and Rani, 2021). Metode seleksi yang digunakan pada penelitian ini adalah seleksi *roulette wheel*. *Roulette wheel selection* metode yang berdasarkan pada nilai *fitness* dalam pemilihan *parent.* Dalam metode ini dapat digambarkan semua kromosom ditempatkan pada roda *roulette* dengan luas bagian setiap individu berdasarkan persentase *fitnessnya.* Sehingga, semakin besar porsi area individu semakin besar juga kemungkinan untuk terpilih sebagai *parent* untuk proses *crossover*. Menurut (Artikel and Chotijah, 2022) langkah-langkah seleksi dengan *Roulette wheel* yaitu sebagai berikut:

1. Menghitung *probabilitas fitness* dari setiap kromosom menggunakan persamaan: $P_i = \frac{f_i}{ft}$ …… (2.11)

   Keterangan:

   Pi = Probabilitas individu ke-i

   Fi = *Fitness* individu ke-i

   Ft = *Fitness* total
2. Menghitung probabilitas komulatif *fitness* setiap kromosom atau penempatan interval nilai masing-masing kromosom.
3. Memutar roda *roulette* sebanyak jumlah individu dalam populasi dan membangkitkan nilai random dengan rentang nilai [0-1]. Jika R[k]<C[i], maka kromosom ke i terpilih sebagai induk.
4. Populasi baru hasil seleksi yang akan digunakan sebagai *parent* dalam proses *crossover*.

d. *Crossover* (Kawin Silang)

*Crossover* merupakan proses untuk menambah keberagaman *string* dalam suatu populasi. Operator *crossover* memiliki peran yang paling penting dalam Algoritma Genetika karena didalamnya terjadi proses persilangan gen antara dua individu *(parent*) yang menghasilkan dua individu baru *(offspring)* pada generasi berikutnya (Setiawan, Herwindiati and Sutrisno, 2019).

*Crossover* menghasilkan dua kromosom anak dengan cara saling bertukar informasi dari dua induk kromosom. Proses ini akan memilih dua kromosom induk yang memiliki nilai *fitness* terbaik kemudian akan mengalami proses kawin silang atau *crossover* secara acak dan akan menghasilkan kromosom anak. Metode *crossover* ada beberapa macam yaitu *crossover* banyak titik, *crossover* satu titik, *crossover* aritmatik, dan *crossover* untuk representasi kromosom permutasi (Ramadhania and Rani, 2021).

Pada penelitian ini menggunakan *arithmetic crossover*, karena metode ini berupa bilangan *real*. Rumus perhitungan *arithmetic crossover* adalah sebagai berikut:

*offspring* 1(x1) = *a* * *parent*1(x1) + (1 - a) * *parent2(x2)*……. (2.12)

*offspring* 2(x2) = (1 - a) * *parent*1(x1) + a * *parent2(x2)*

Keterangan:

*offspring* 1= Keturuna pertama

*offspring* 2= Keturunan kedua

a = *Crossover rate* dengan nilai 0 sampai 1

*Parent* 1 = Orang tua pertama

*Parent* 2 = Orang tua kedua

e. *Mutation* (Mutasi)

Mutasi merupakan suatu proses untuk memperoleh individu baru yang lebih baik dengan cara memodifikasi satu atau lebih gen yang ada pada suatu individu. Mutasi berfungsi untuk mengelola keberagaman dalam populasi dan mencegah pemusatan data (konvergensi) yang terlalu cepat. Algoritma akan terhenti ketika populasi telah memusat (konvergen) dan tidak menghasilkan keturunan yang tidak beda jauh dari generasi sebelumnya. Hal tersebut berarti masalah yang telah didefinisikan telah menemukan solusi (Rozi, Firdausi and Rahmadhany, 2021).

Mutasi akan menghasilkan populasi menjadi bervariasi dengan menukar gen yang hilang dari populasi selama proses seleksi serta menyimpan gen yang tidak ada dalam populasi awal (Ramadhania and Rani, 2021).

Proses mutasi memerlukan *mutation rate* dengan nilai yang telah ditentukan dari awal yang memiliki rentang nilai antara 0 sampai 1 (Setiawan, Herwindiati and Sutrisno, 2019). Sehingga pada penelitian ini menggunakan *random mutation*. Nilai *offspring* dibangkitkan dengan menggunakan rumus sebagai berikut:

$x_i' = x_i + r(max_i - min_i)$ ……(2.13)

r memiliki rentang nilai antara -0,1 sampai 0,1.