BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2. 1 Tinjauan Pustaka

Pada penelitian ini menggunakan beberapa referensi terkait dengan metode KNN dan optimasi menggunakan algoritma genetika. Hal ini berfungsi sebagai pedoman sekaligus pembanding dengan penelitian terdahulu terhadap penelitian yang akan dilakukan. Referensi tersebut diantaranya sebagai berikut:

Fahlevi (2020) menerapkan algoritma genetika untuk mengoptimalkan parameter K pada *Modified K-Nearest Neighbour* (MKNN) pada kasus penerima kasus beras sejahtera. Diperoleh akurasi terbaik sebaik sebesar 88% dengan nilai K=15, K=7, dan K=9 dimana probabilitas *crossover* yang digunakan sebesar 0,7 atau 0,8 dan probabilitas mutasi sebesar 0,3 atau 0,2.

Renalia (2019) memanfaatkan algoritma genetika untuk mengoptimalkan penentuan *centroid* pada K-Means, sehingga didapatkan hasil pengelompokkan mahasiswa TI STMIK Akakom angkatan 2015/2016 menjadi 3 klaster. Akan tetapi, hasil akhir yang diperoleh belum mampu secara baik mengoptimalkan K-Means dengan nilai *silhouette coefficient* cukup jauh dari 1. Hal ini disebabkan oleh data yang digunakan kurang heterogen dan jumlahnya terlalu sedikit.

Mahardia, dkk (2018) menggunakan algoritma *particle swarm optimization* (PSO) untuk mengoptimalkan parameter nilai K pada KNN dalam kasus pengendalian hama tanaman jeruk. Penelitian tersebut membuktikan bahwa metode PSO-KNN mampu meningkatkan akurasi mencapai 96,25%. Akurasi tersebut lebih

tinggi dibandingkan dengan hanya menggunakan metode KNN saja, yaitu sebesar 90%. Sehingga PSO dapat memperbaiki kekurangan pada algoritma KNN.

Astuti, dkk (2017) menerapkan algoritma genetika untuk mengoptimalkan parameter nilai K pada MKNN dimana nilai fitness diperoleh dari perhitungan rata-rata validitas semua data latih terhadap nilai K. Hasil yang didapat dari proses optimasi pada kasus deteksi penyakit pada kucing ini menghasilkan tingkat akurasi hingga 100% ketika nilai K=1.

Lesmono, dkk (2017) memanfaatkan algoritma genetika untuk mengoptimalkan penentuan parameter nilai K pada KNN sehingga akurasi pada dataset iris meningkat hingga mencapai akurasi tertinggi sebesar 99%. Akan tetapi, untuk menghasilkan nilai K yang optimal algoritma genetika membutuhkan waktu yang cukup lama sehingga proses klasifikasi berjalan cukup lambat.

Penelitian ini menggunakan algoritma genetika untuk melakukan optimasi pada penentuan parameter K algoritma KNN yang kemudian akan digunakan untuk klasifikasi penentuan obat pada pasien. Perbandingan antara beberapa penelitian terdahulu disajikan pada tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

No	Nama Penulis	Objek	Algortima	Metode Optimasi	Hasil
1.	Rizki Fahlevi (2020)	Penerima beras sejahrera	MKKN	Algoritma Genetika	Metode algoritma genetika dapat digunakan pada MKNN untuk menentukan nilai K yang optimal dengan menghasilkan akurasi tertinggi sebesar 88%
2.	Elsa Renalia (2019)	Mahasiswa STMIK Akakom jurusan TI angkatan 2015/2016.	K-Means	Algoritma Genetika	hasil yang diperoleh belum mampu mengoptimasi K-Means (silhouette coefficient < 1) dikarenakan jumlah data kurang banyak dan heterogenitas data juga terlalu sedikit.
3.	Mahardika dkk (2018)	Hama pada Tanaman Jeruk.	KNN	PSO	Metode PSO mampu memperbaiki kekurangan pada algoritma KNN yang terbukti dengan peningkatan akurasi ketika KNN digabungkan dengan PSO.
4.	Fitri Dwi Astuti, dkk (2017)	Pasien di klinik hewan kota kediri	MKNN	Algoritma Genetika	Algoritma genetika dapat digunakan untuk optimasi nilai K pada metode MKNN dengan nilai optimal pada K=1 menghasilkan akurasi 100% dalam permasalahan deteksi penyakit kucing.
5.	Lesmono dan Praba (2017)	Dataset public yaitu Tic-Tac-Toe Endgame, iris dan Image Segmentation	KNN	Algoritma genetika	Algoritma genetika mampu meningkatkan akurasi hingga 99% pada algoritma KNN ketika dilakukan optimasi pada nilai K.
6.	Ardina Surya G. (2022)	Dataset Drug Tahun 2018	KNN	Algoritma Gentetika	Algoritma genetika mampu membantu menentukan parameter K paling optimal yang akan digunakan pada KNN

2. 2Dasar Teori

2.1.1 Data Mining

Data mining merupakan proses menggali atau menambang (*mining*) data berukuran besar pada data *warehouse* menggunakan kecerdasan buatan,matematika dan statistik sehingga menghasilkan pengetahuan baru. (Jollyta, Ramadan, & Zarlis, 2020) Teknologi data mining diharapkan bisa menjadi penghubung antara data dengan penggunanya.

Secara garis besar kegunaan data mining dibagi menjadi dua, yaitu kegunaan deskriptif yang berfungsi untuk mencari pola tertentu dari suatu data sehingga dapat digunakan untuk menemukan karakteristik yang mudah dipahami oleh manusia. Sedangkan untuk kegunaan prediktif berfungsi untuk menemukan model pengetahuan sehingga dapat digunakan untuk melakukan.

2.1.2 Metode Klasifikasi Data Mining

Metode klasifikasi merupakan teknik yang didasarkan pada atribut dari kelompok yang sudah didefinisikan. Sehingga didapatkan suatu aturan yang digunakan untuk melakukan klasifikasi pada data baru dengan cara memanipulasi data yang sudah ada dan sudah diklasifikasi . (Novriansyah & Nurcahyo, 2015).

Metode ini termasuk ke dalam kelompok *supervised learning* yang setiap *item* datanya memiliki label atau kelas yang dipengaruhi atribut. Tipe data yang cocok digunakan pada metode klasifikasi yaiu biner atau nominal sedangkan untuk tipe data ordinal kurang cocok sebab pada metode ini menggunakan pendekatan secara implisit . (Novriansyah & Nurcahyo, 2015).

2.1.3 K-Nearest Neighbor

Algoritma KNN didasarkan pada pembelajaran dengan analogi yaitu membandingkan data uji yang diberikan dengan data latih yang serupa. Dimana data latih dideskripsikan oleh n-atribut yang kemudian setiap *record* pada data latih disimpan dalam n-dimensi. Sehingga, ketika diberikan suatu *record* data yang belum diketahui maka KNN akan mencari pola untuk K data latih yang paling dekat dengan *record* yang belum diketahui. (Han, Kamber, & Pei, 2012)

Menurut (Suyanto, 2017) ada beberapa hal menarik pada algoritma KNN yaitu mudah diimplementasikan hanya menggunakan cara yang sederhana dengan menentukan satu parameter K dan algoritma KNN bekerja secara lokal dengan hanya memperhitungkan sejauh K data. Namun, disisi lain KNN juga memiliki kelemahan yaitu sangat sensitif terhadap *noise* ataupun *outlier* pada data. Selain itu, pada algoritma ini kesulitan menentukan parameter K dalam proses pelatihan. Parameter K yang optimal hanya bisa ditemukan secara empiris berdasarkan beberapa kali percobaan terhadap pola-pola representatif dengan jumlah yang memadai (Suyanto,2017).

Langkah-langkah algoritma KNN yaitu:

1. Memasukkan data latih dan data uji
2. Menentukan nilai K
3. Menghitung jarak *eucledian* setiap data latih terhadap data uji menggunakan rumus pada persamaan 2.1.

$$(x, y) = \sqrt{\sum^{n}{}_{i=1}(xi_{training} - yi_{testing})^2} \quad \text{.....(2.1)}$$

Keterangan:

d(x,y): jarak antara data latih dengan data uji ,

n : jumlah data latih,

x : data latih,

y : data uji.

4. Mengurutkan hasil perhitunagan jarak mulai dari terkecil ke yang terbesar.
5. Mengumpulkan atau mengambil sejumlah data sesuai nilai K yang telah ditentukan pada langkah ke-2.
6. Menentukan hasil dari pengambilan data berdasarkan tetangga terdekat pada langkah ke-4 dapat diklasifikasian berdasarkan kategori yang telah ditentukan.

2.1.4 Algoritma Genetika

Algoritma genetika memiliki prinsip utama untuk meniru proses seleksi alam dimana setiap individu akan bersaing untuk bertahan hidup dalam melakukan reproduksi untuk menghasilkan keturunan. Pada algoritma genetika hanya individu-individu yang "*fit*" yang akan memiliki peluang untuk hidup dan sebaliknya individu yang kurang "*fit*" akan mati (prinsip *survival of the fittest*).

Pada proses seleksi alam akan "dilahirkan" individu baru yang lebih "*fit*" dari *parent*-nya melalui proses persilangan (*crossover*) dan mutasi. Pada algoritma genetika proses seleksi dan reproduksi (persilangan dan mutasi) akan terus berulang sampai dihasilkan individu baru yang "*fit*" (Arkeman, Herdiyeni, Hermadi, & Laxmi, 2014).

Tabel perbandingan istilah-istilah yang digunakan pada proses evolusi biologi dengan proses pada algoritma genetika disajikan pada tabel 2.2 (Goldberg, 1989)

Tabel 2. 2 Perbandingan Istilah-Istilah Pada Algoritma Genetika

No	Biologi	Algoritma Genetika
1.	Individu	Kandidat solusi
2.	Populasi	Kumpulan dari kandidat solusi
3.	Fitness	Kualitas dari solusi yang dihasilkan
4.	Kromosom	String yang mewakili setiap individu
5.	Gen	Karakter
6.	Allele	Kualitas dari karakter

Langkah-langkah algortima genetika yang digunakan pada penelitian ini dijelaskan pada uraian berikut:

1. Populasi Awal

Populasi awal merupakan sekumpulan kromosom (kandidat solusi) yang secara acak dibangkitkan. Sebelum membangkitkan populasi awal harus diperhatikan jumlah populasi yang akan digunakan. Karena apabila jumlah populasi terlalu sedikit maka populasi akan mencapai konvergen terlalu cepat yang mengakibatkan terjadinya lokal optima. sebaliknya jika jumlah populasi terlalu banyak maka akan membutuhkan waktu komputasi yang lama dan proses perhitungan yang kompleks (Kismanti, 2016).

kromosom-kromosom pada populasi awal tersebut harus dilakukan pengkodean (*encoding*) ke dalam bentuk gen sehingga bisa diproses dalam algoritma genetika. Terdapat beberapa pengkodean pada algoritma genetika, yaitu:

a. Pengkodean bilangan real : nilai berada pada interval [0 1]

b. Pengkodean diskrit desimal : nilai berupa bilang bulat pada interval [0 9]

c. Pengkodean biner : nilai berupa bilangan biner 1 atau 0.

2. Evaluasi Fitness

Evaluasi fitness bertujuan untuk mengukur kualitas dari solusi yang didapatkan kemudian memungkinkan untuk dilakukan perbandingan antar tiap solusi (Michalewicz, 1996). Dimana terdapat beberapa cara untuk mendapatkan nilai fitness yang disesuaikan dengan permasalahan yang akan diselesaikan. Pada penelitian ini nilai fitness diperoleh berdasarkan nilai validitas. Langkah-langkah mencari nilai validitas yaitu:

a. *Euclidian Distance* Antar Data Latih

Pencarian *euclidian distance* antar data latih ini dilakukan untuk mengetahui kedekatan dan kemiripan antar dua titik pada data latih. Dimana semakin minimum jarak antar dua data maka akan memiliki tingkat kemiripan yang tinggi dan *euclidian distance* bisa dikatakan baik. Persamaan 2.1 digunakan untuk mencari *euclidian distance*:

$$(x, y) = \sqrt{\sum^{n}_{l=1}(xi_{training\ a} - yi_{training\ a+1})^2} \quad \text{.....} \quad (2.2)$$

Keterangan:

d(x,y) : jarak antar data latih

n : jumlah data latih,

$xi_{training\ a}$: data latih data ke a ,

$yi_{training\ a+1}$: data latih ke a+1

b. Mencari Validitas Setiap Data Latih Berdasarkan Similaritas

Similaritas merupakan fungsi yang digunakan untuk mengukur kemiripan antara data kasus baru dengan data kasus lama (Hendra & Kusumadewi, 2015). Semakin tinggi tingkat kemiripannya maka akan semakin besar peluang kesamaan solusi antar kedua data. Persamaan 2.2 digunakan untuk mencari nilai similaritas:

$S\ a, b\begin{cases}1 & a=b \\ 0 & a\neq b\end{cases}$ ……..…(2.3)

Keterangan:

S: Similaritas

a: data kasus lama

b: data kasus baru

Berdasarkan persamaan 2.2 maka dapat diambil contoh sebagai berikut:

a. Similaritas data A (12) dan data B (5)

   S(A,B) = (12≠5) ↔ S(A,B) = 0

b. Similaritas data A (12) dan data C (12)

   S(A,C) = (12=12) ↔ S(A,C) = 1

Selanjutnya nilai similaritas digunakan untuk mencari nilai validitas, dimana besarnya nilai validitas setiap data bergantung pada data tetangga terdekatnya. Rumus validitas yaitu pada persamaan 3 di bawah:

$validitas_{(i)} = \sum_{i=1}^{k} S(lbl_x, lbl\ Ni_x)$ ……(2.4)

Keterangan:

K = jumlah keseluruhan tetangga terdekat

Lbl (x) = kelas x

Lbl Ni = label kelas titik paling dekat x

c. Mencari nilai fitness

Nilai fitness merupakan ukuran kualitas dari suatu individu yang dari fungsi fitness. Suatu individu yang berkualitas tinggi memiliki nilai fitness yang paling besar. Pada penelitian ini nilai fitness dicari berdasarkan nilai rata-rata dari nilai validitas yang telah dicari pada langkah sebelumnya. Jika dituliskan ke dalam bentuk persamaan maka dapat dilihat pada persamaan 3.4.

Fitness = $\bar{x}validitas_{(i)}$……..…(2.5)

Keterangan:

$\bar{x}$: rata-rata

$validitas_{(i)}$: nilai validitas data ke-i

3. Menentukan *Global maximum* (Gmax)

Global maximum merupakan nilai keseluruhan terbesar dari suatu fungsi. Sehingga dalam penelitian ini penentukan *global maximum* didasarkan pada kromosom yang memiliki nilai fitness paling tinggi.

4. Representasi kromosom

Pengkodean merupakan langkah dalam merepresentasikan masalah dalam bentuk kromosom. Dalam penelitian ini, setiap gen yang terdapat dalam kromosom direpresentasikan ke dalam bentuk biner melalui pengkodean *binary encoding*, dimana setiap *allele* direpresentasikan dalam bentuk *bit* 0 atau *bit* 1.

5. Seleksi

Seleksi kromosom digunakan untuk menentukan bahwa jumlah kromosom generasi berikutnya akan bergantung pada nilai fitness masing-maisng kromosom yang dibandingkan dengan rata-rata nilai fitness pada populasi tersebut (Arkeman Y. , 2012). Kromosom-krosomom yang memiliki nilai fitness terbaik akan

memiliki peluang lebih besar untuk menjadi induk dan bertahan pada generasi berikutnya. Sebaliknya kromosom dengan nilai fitness yang kecil akan tergantikan dengan kromosom baru yang memiliki fitness lebih besar.

Beberapa teknik yang biasa digunakan untuk melakukan seleksi pada algoritma genetika,yaitu:

a. Mesin Reoulette (Roulette wheel)

Teknik seleksi ini dilakukan dengan menempatkan setiap kromosom pada slot cakram rolet. Dimana besarnya ukuran slot sebanding dengan rasio antara nilai fitness pada suatu kromosom dan total nilai fitness dari keseluruhan kromsom. Kemudian rolet diputar sejumlah ukuran populasi yang telah ditentukan (Goldberg, 1989).

b. Tournament

Teknik seleksi ini dilakukan dengan mengeliminasi dua objek dengan cara dipilih kemudian diadu. Dimana objek yang menang akan melakukan reproduksi sedangkan objek yang kalah akan musnah (Rich, 1995).

c. Elitism

Teknik seleksi ini dilakukan dengan mengumpulkan dalam satu penampungan dari individu baik populasi dan *offspring*. Kemudian memilih individu-individu tersebut untuk menjadi generasi selanjutnya berdasarkan nilai fitness tertinggi. Sehingga pada teknik seleksi ini hanya individu terbaik yang akan bisa lolos pada generasi selanjutnya (Mahmudy, 2013)

6. *Crossover* (Penyilangan)

Operator algoritma ini bekerja untuk melakukan penyilangan pada sepasang kromosom induk sehingga menghasilkan dua kromosom anak. Penyilangan dilakukan dengan cara menukarkan sejumlah gen yang dimiliki krosomsom induk.

Crossover probability (CP) merupakan rasio antara jumlah kromosom yang diharapkan melakukan *crossover* dalam satu generasi dengan jumlah keseluruhan kromosom dalam populasi. Dalam algoritma genetika, *crossover* merupakan operator primer. Sehingga nilai CP yang digunakan juga cukup tinggi antara 0,6 sampai 1. (Arkeman Y. , 2012).

Jika CP tinggi maka akan meningkatkan kemungkinan algoritma genetika untuk menjelajahi ruang pencarian sehingga solusi optimum akan semakin cepat ditemukan. Namun, penentuan nilai CP yang terlalu tinggi juga akan menyebabkan waktu pencarian pada daerah yang kurang menjanjikan (*unpromising region*) menjadi terbuang (Arkeman Y. , 2012).

Beberapa teknik yang biasa digunakan untuk melakukan *crossover* pada algoritma genetika,yaitu:

a. *One-Cut Point Crossover*

Teknik *crossover* ini cocok digunakan pada representasi kromosom biner (Arkeman Y. , 2012). Teknik ini dilakukan dengan cara memilih satu bit pada kromosom kemudian menukar masing masing gen sebelah kanan titik *crossover*. Dalam menentukan titik potong *crossover* dilakukan pembangkitan bilangan bulat secara acak (Syarif, 2014).

b. *Two-Cut Point Crossover*

Teknik ini dilakukan dengan cara memilih dua bit pada kromosom *parent* pertama kemudian ditukar dengan dua bit kromsom pada *parent* kedua pada posisi yang sama. *Parent* ditukarkan secara random pada *substring*.

c. Order crossover

Teknik ini dilakukan dengan cara membangkitkan dua bilangan random. Kemudian gen yang berada diantara kedua bilangan random akan disalin ke *offspring* dalam posisi yang sama (Davis, 1985)

7. Mutasi

Operator mutasi dilakukan dengan mengubah nilai gen dalam suatu kromosom. Operator ini bertujuan untuk mendapatkan kromosom-kromosom baru sebagai kandidat solusi untuk generasi selanjutnya dengan nilai fitness lebih tinggi hingga menghasilkan solusi optimum (Murniati, 2009)

Mutation probability (MP) merupakan rasio antara jumlah kromosom yang diharapkan melakukan mutasi dalam satu generasi dengan jumlah keseluruhan kromosom dalam populasi. Dalam algoritma genetika, mutasi merupakan operator sekunder. Sehingga nilai MP yang digunakan juga cukup rendah antara 0,001 sampai 0,2 (Arkeman Y. , 2012).

Ketika MP rendah maka akan semakin kecil peluang munculnya gen-gen baru. Namun, disisi lain gen baru dibutuhkan untuk mendapatkan solusi optimum. Sebaliknya, jika MP terlalu tinggi maka akan mengakibatkan munculnya banyak

mutan yang akan menghilangkan karakteristik dari kromosom induk pada generasi selanjutnya (Gen & Cheng, 1997).

a. Swapping Mutation

Teknik mutasi ini diawali dengan memilih dua bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua dalam probabilitas tertentu (Suyanto, 2005).

b. Flip Mutation

Teknik mutasi ini dilakukan dengan cara menukarkan atau membalikkan gen yang bernilai 1 menjadi 0. Begitu sebaliknya gen yang bernilai 0 menjadi 1 (Syarif, 2014)

c. Random Mutation

Teknik mutasi ini dilakukan dengan cara memilih satu kromosom induk secara random dari populasi. Kemudian dilakukan penambahan atau pengurangan nilai kromosom pada gen terpilih dengan bilangan random terkecil (Mahmudy, 2013).

8. Populasi Baru

Pembentukkan populasi baru didasarkan pada *offspring* baru yang dihasilkan dari operasi mutasi ditambah dengan individu terbaik. Setelah dihasilkan populasi baru, maka selanjutnya mengulangi langkah-langkah evaluasi nilai fitness, seleksi, *crossover*, dan mutasi hingga menghasilkan nilai optimal.

2.1.5 Confusion Matrix

Confusion matrix digunakan untuk melakukan evaluasi kinerja pada metode klasifikasi dengan menganalisis tingkat akurasi dari *classifier* dalam mengenali *tuple* dari kelas yang berbeda. Ada beberapa istilah yang digunakan dalam *confusion matrix* yaitu TP (*True Positive*) dan TN (*True Negative*) memberikan informasi jika *classifier* benar sedangkan FP (*False Positive*) dan FN (*False Negative*) memberikan informasi ketika *classifier* salah. (Han, Kamber, & Pei, 2012)

Confusion matrix merupakan salah metode yang digunakan untuk mengukur performa dari suatu model klasifikasi yang telah dibuat, dimana *output* dapat berupa dua kelas atau banyak kelas. Confusion matrix menggunakan tabel yang berisi empat kombinasi dari nilai aktual dan nilai prediksi dari hasil pengujian. Tabel tersebut dapat dilihat pada gambar 2.2.

	Predict Class		
	Yes	No	
Yes	TP	FN	
No	FP	TN	**Actua Class**
Total	P'	N'	

Gambar 2. 1 Confusion matrix dua kelas

Berdasarkan gambar 2.2 pada *confusion matrix* terdapat *predict class* dan *actual class*. Dimana *predict class* merupakan keluaran dari program yang diberi nilai positif dan negatif sedangkan *actual class* merupakan nilai sebenarnya yang

diberi nilai *true* dan *false*. Berikut penjelasan lengkap untuk *actual class* dan *predict class* pada *confusion matrix*.

1. True Positive (TP) : Model memprediksi positif dan secara aktual itu benar.
2. True Negative (TN) : Model memprediksi negatif dan secara aktual itu benar.
3. False Positive (FP):Model memprediksi positif namun secara aktual itu salah.
4. False Negative (FN):Model memprediksi negatif namun secara aktual itu salah.

Dari *confusion matrix* maka dapat dihitung nilai akurasi yang merepresentasikan seberapa akurat model klasifikasi yang telah dibuat dalam melakukan pengklasifikasian secara benar menggunakan persamaan berikut (Anggreany, 2022).

$$\text{Akurasi } (2\ kelas) = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} . 100\% \text{…..……(2.6)}$$

$$\text{Akurasi}(> 2 kelas) = \frac{\text{TP}}{\text{Total Data}} . 100\% \text{ … ….…….…...……(2.7)}$$