Optimasi Klasifikasi Batubara Berdasarkan Jenis Kalori dengan menggunakan *Genetic Modified* K-*Nearest Neighbor* (GMK-NN) (Studi Kasus: PT Jasa Mutu Mineral Indonesia Samarinda, Kalimantan Timur)

Optimization of Coal Classification Based on Calorie using Genetic Modified K-Nearest Neighbor (GMK-NN) (*Case Study:* PT Jasa Mutu Mineral Indonesia Samarinda, Kalimantan Timur)

Nanang Wahyudi[1], Sri Wahyuningsih[2], dan Fidia Deny Tisna Amijaya[3]
[1]Laboratorium Statistika Komputasi FMIPA Universitas Mulawarman
[2]Laboratorium Statistika Terapan FMIPA Universitas Mulawarman
[3]Laboratorium Matematika Komputasi FMIPA Universitas Mulawarman
E-mail: nanangwahyudi100@gmail.com

Abstract

The K-Nearest Neighbor (K-NN) method is one of the oldest and most popular Nearest Neighbor-based methods. The researchers developed several methods to improve the performance of the K-NN algorithm by using the Genetic Modified K-Nearest Neighbor (GMK-NN) algorithm. This method combines the genetic algorithm and the K-NN algorithm in determining the optimal K value used in the classification prediction. The GMK-NN algorithm will greatly facilitate the examination of coal classification in the laboratory without having to do a lot of chemical and physics testing that takes a long time only with the data already available. In this research, K value optimization is done to predict the classification of coal based on calories owned by PT Jasa Mutu Mineral Indonesia in 2017. Based on the research, using the proportion of training and testing data 90:10, 80:20 and 70:30 obtained the value of K the most optimal is at K = 1. The highest prediction accuracy was obtained by using 90:10 proportion data which is 100%, then with the proportion of 80:20 data obtained prediction accuracy of 91.67% and with the proportion of 70:30 data obtained prediction accuracy of 94.44%.

Keywords*: Classification, genetic algorithm, K-Nearest Neighbor*

Pendahuluan

Klasifikasi adalah salah satu tugas dari *data mining* yang bertujuan untuk memprediksi label kategori benda yang tidak diketahui sebelumnya, dalam membedakan antara objek yang satu dengan yang lainnya berdasarkan atribut atau variabel.Berdasarkan cara pelatihan, algoritma-algoritma klasifikasi dapat dibagi menjadi dua macam, yaitu *eager learner* dan *lazy learner*. Salah satu algoritma yang masuk ke dalam pelatihan *lazy learner* adalah Algoritma K-*Nearest Neighbor* (K-NN) (Prasetyo, 2014).

Metode K-NN menjadi salah satu metode berbasis NN yang paling tua dan populer. Nilai K yang digunakan pada metode K-NN menyatakan jumlah tetangga terdekat yang dilibatkan dalam penentuan prediksi label kelas pada data *testing*.Ada beberapa hal yang memengaruhi kinerja akurasi K-NN, di antaranya adalah pemilihan nilai K.

Karena dirasa masih memiliki beberapa kekurangan maka para peneliti mengembangkan beberapa metode untuk meningkatkan kinerja algoritma K-NN, salah satunya adalah algoritma *Modified* K-*Nearest Neighbor* (MK-NN) (Parvin. dkk, 2010). Tujuan dari algoritma MK-NN adalah untuk meningkatkan akurasi dari K-NN, dengan menambahkan fungsi Validitas dan *Weight Voting*. Namun, algoritma MK-NN juga masih memiliki kekurangan yaitu komputasi yang kompleks dan nilai K yang masih bias. Untuk mengatasi masalah ini dilakukan penelitian untuk masalah optimasi nilai K dengan menambah algoritma genetika (*genetic algorithm*) dalam penentuan nilai K optimal. Algoritma genetika selain berguna untuk menentukan nilai K secara otomatis juga dapat meningkatkan nilai kinerja akurasi dan dapat mengurangi kompleksitas dalam komputasi.

Penelitian sebelumnya yang dilakukan oleh Mutrofin (2015) yang berjudul "Optimasi Teknik Klasifikasi MK-NN Menggunakan Algoritma Genetika", peneliti melakukan optimasi nilai K dengan *Genetic Algorithm* (GA) yang digunakan pada K-NN menggunakan dataset *Iris* dan *Wine* dengan GMK-NN mendapatkan K optimal sebesar 2 dengan nilai akurasi 100% untuk data *training* sebesar 70% dan 90%.

Data klasifikasi biasanya memiliki kriteria-kriteria tertentu agar suatu sampel data dapat masuk ke dalam kelas tertentu. Salah satu data yang memiliki klasifikasi adalah data jenis batubara. Batubara merupakan komoditas energi yang semakin banyak dieksplorasi dan dieksploitasi untuk pemenuhan kebutuhan energi masyarakat dunia. Penafsiran hasil pemeriksaan laboratorium untuk mengetahui klasifikasi batubara tidak dapat menggunakan satu jenis hasil

pemeriksaan saja, tetapi menggunakan gabungan beberapa hasil pemeriksaan. Hal itu disebabkan sifat hasil pemeriksaan laboratorium pada batubara menjadi tidak spesifik (Arif, 2014). Dengan menggunakan algoritma GMK-NN dengan data-data yang dimiliki akan sangat memudahkan pemeriksaan klasifikasi batubara di laboratorium tanpa harus melakukan banyak pengujian kimia dan fisika yang membutuhkan waktu yang lama.

Berdasarkan uraian di atas, penulis tertarik untuk mengkaji analisis klasifikasi algoritma yang termasuk ke dalam kategori *lazy learner* dengan mengambil studi kasus klasifikasi batubara di PT Jasa Mutu Mineral Indonesia.

Data Mining

Data mining merupakan bidang multi-disiplin yang melibatkan pembelajaran mesin, statistik, *database*, kecerdasan buatan, pencarian informasi dan visualisasi. *Data mining* juga disebut *Knowledge Discovery In Database* (KDD). Hal ini biasanya didefinisikan sebagai proses menemukan pola yang berguna atau pengetahuan dari sumber data, misalnya *database*, teks, gambar, web, dan lain-lain.

Konsep Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untukmemasukannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalamklasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan modelsebagai prototipe untuk disimpan sebagai memori dan penggunaan model tersebutuntuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agardiketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya (Prasetyo, 2012).

Konsep Kedekatan

Metode klasifikasi seperti K-NN dan metode-metode *clustering* biasanya menggunakan suatu kuantitas yang disebut kedekatan atau *proximity*. Ada dua jenis kedekatan yaitu kemiripan (*similarity*) atau ketidakmiripan (*dissimilarity*).

Ukuran ketidakmiripan yang paling umum digunakan adalah jarak *Euclidean* yang diformulasikan oleh persamaan berikut:

$$d(x_a, y_b) = \sqrt{\sum_{i=1}^{r} (x_{ai} - y_{bi})^2} \qquad (1)$$

di mana:

x_a = nilai ke-*a* dari data *x*
y_b = nilai ke-*b* dari data *y*
x_{ai} = nilai ke-*a* variabel ke-*i* dari data *x*
y_{bi} = nilai ke-*b* variabel ke-*i* dari data *y*
r = jumlah variabel
d = jarak *Euclidean*

Selain itu pengukuran kemiripan yang diusulkan oleh Gower (1971) menyatakan jika nilai yang dicari adalah biner, maka:

$$s(x,y) = \begin{cases} 1, & \text{jika } x = y \\ 0, & \text{lainnya} \end{cases} \qquad (2)$$

Persamaan (2) menyatakan bahwa jika *x* dan *y* yang bertipe nominal atau ordinal maka $s(x, y)$ bernilai 1 jika mempunyai nilai yang sama dan sebaliknya maka akan bernilai 0.

Normalisasi Data

Variabel yang memiliki nilai yang paling besar memiliki pengaruh yang lebih kecil dalam melakukan prediksi klasifikasi daripada variabel yang memiliki nilai yang kecil. Untuk mengatasi masalah tersebut dapat digunakan teknik normalisasi variabel sehingga semua variabel akan berada pada jangkauan yang sama.Cara untuk menentukan nilai normalisasi adalah dengan menghitung nilai rata-rata dan variansi seperti dalam persamaan berikut:

$$\bar{x}_j = \frac{1}{N} \times \sum_{i=1}^{N} x_{ij} \qquad (3)$$

$$\sigma_j^2 = \frac{1}{N-1} \times \sum_{i=1}^{N} (x_{ij} - \bar{x}_j)^2 \qquad (4)$$

$$\hat{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j} \qquad (5)$$

di mana:

N = banyak data
x_{ij} = data ke-*i* pada variabel ke-*j* dimana $j = 1,2,\ldots,r$
$\bar{x}_j$ = rata-rata pada variabel ke-*j*
σ_j^2 = variansi
σ_j = standar deviasi
$\hat{x}_{ij}$ = normalisasi data ke-*i* variabel ke-*j* (Prasetyo, 2014).

Modified K-*Nearest Neighbor* (MK-NN)

Metode K-NN menjadi salah satu metode berbasis NN yang paling tua dan populer. Ada beberapa hal yang memengaruhi kinerja akurasi K-NN, di antaranya adalah pemilihan nilai K. Jika K terlalu kecil maka berakibat hasil prediksi yang didapat bisa sensitif terhadap keberadaan *noise*. Di sisi lain, jika K terlalu besar maka tetangga terdekat yang terpilih mungkin terlalu banyak dari kelas lain yang sebenarnya tidak relevan karena jarak yang terlalu jauh.

Karena dirasa masih memiliki beberapa kekurangan maka para peneliti mengembangkan beberapa metode untuk meningkatkan kinerja algoritma K-NN, salah satunya adalah algoritma *Modified* K-*Nearest Neighbor* (MK-NN) (Parvin.

dkk, 2010). Tujuan dari algoritma MK-NN adalah untuk meningkatkan akurasi dari K-NN, dengan menambahkan fungsi validitas dan *Weight Voting*. Nilai validitas diperoleh dengan melakukan perhitungan jarak *Euclidean* pada masing-masing data *training* dan menghitung *similarity*-nya. Algoritma MK-NN memiliki kelebihan yaitu mampu mengatasi akurasi rendah dari *Weighted* K-*Nearest Neighbor* (WK-NN), lebih stabil dan kuat. Namun, algoritma MK-NN juga memiliki kekurangan yaitu komputasi yang masih kompleks. Adapun rumus fungsi validitas dapat dilihat pada Persamaan (6).

$$validitas(a) = \frac{1}{K}\sum_{i=1}^{K} s(target(x_a), target(n_i(x_b))) \quad (6)$$

di mana:

$validitas(a)$ = validitas data *training* ke-*a*

K = jumlah tetangga antar data

s = s*imilarity* (nilai 1 = terbaik)

$target(x_a)$ = label kelas data *training*

$target(n_i(x_b))$ = label kelas jarak terdekat pada data *training*

Nilai fungsi validitas diperoleh dari perhitungan similaritas antara *target* data *trainingx* dengan *target* data *training* tetangganya $n_i(x)$ yang ditentukan dengan menggunakan rumus *similarity* pada Persamaan (7).

$$s(x_a, x_b) = \begin{cases} 1, x_a = x_b \\ 0, x_a \neq x_b \end{cases} \quad (7)$$

Sedangkan rumus bobot (*weight voting*) dapat dilihat pada persamaan berikut:

$$w(x_a, y_b) = validitas(a) \times \frac{1}{d(x_a, y_b) + \alpha} \quad (8)$$

di mana:

w = Bobot antara data *training* dan data *testing*

$validitas(a)$ = validitas data *training*

$d(x_a, y_b)$ = jarak antar data *training*

α = 0,5 (parameter *smoothing* yang di-tentukan oleh peneliti)

Setelah menghitung bobot berdasarkan nilai validitas, dengan melanjutkan perhitungan *voting* berdasarkan bobot tiap kelas dari data uji. Adapun rumus *voting* dapat dilihat pada persamaan (2.20).

$$vote(Kelas) = \sum_{i=1}^{n} w(x_a, y_b)_n \quad (9)$$

di mana:

$vote(Kelas)$ = *voting* kelas berdasarkan bobot data *testing* dan data *training*pada kelas yang sama

$w(x_a, y_b)_n$ = nilai *weight voting* data *training* dengan data *testing* ke-*i* padakelas yang sama

Dalam perhitungan *voting*, fungsi *voting* ini adalah untuk menentukan data uji itu lebih cenderung ke kelas apa, *voting* sangat membantu ketika sebuah data uji memiliki kecenderungan pada kelas yang lebih dari satu kelas. Setelah melakukan perhitungan, *voting* diurutkan dari nilai terbesar ke nilai terkecil dan *voting* dengan nilai terbesar akan menjadi kelas yang terpilih (Mutrofin, 2015).

Pengukuran Kinerja Klasifikasi

Sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar, tetapi tidak dipungkiri bahwa kinerja suatu sistem tidak bisa 100% benar sehingga sebuah sistem klasifikasi juga harus diukur kinerjanya (Rodiyansyah, 2013).Untuk menghitung persentase akurasi digunakan persamaan:

$$\text{Akurasi} = \frac{\text{Jumlah data yang diprediksi benar}}{\text{jumlah prediksi yang dilakukan}} \times 100\% \quad (9)$$

Algoritma Genetika

Algoritma Genetika merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin. Metode optimasi dikembangkan oleh John Holland sekitar tahun 1960-an dan dipopulerkan oleh salah seorang mahasiswanya David Goldberg pada tahun 1980-an. Proses pencarian penyelesaian atau proses terpilihnya sebuah penyelesaian dalam algoritma ini berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam proses evolusi.Pada mulanya, populasi awal dibangkitkan secara acak sesuai dengan representasi masalah yang akan dikembangkan. Selanjutnya, operator-operator genetika akan menggabungkan informasi genetis dari unsur-unsur populasi untuk membentuk populasi generasi berikutnya. Setiap kromosom mempunyai nilai *fitness* yang setara dengan nilai penyelesaian masalah. Pada generasi berikutnya, nilai *fitness* kromosom sebagai representasi dari penyelesaian masalah, diharapkan bertambah semakin bagus.

Berikut ini adalah struktur dasar algoritma genetika:

1. Inisialisasi Populasi

   Proses inisialisasi populasi adalah proses membangkitkan kromosom secara acak sebanyak ukuran populasi. Pengkodean merupakan bagian penting dalam tahapan inisialisasi. Proses ini diperlukan dalam kaitannya dengan peranan kromosom sebagai representasi penyelesaian masalah.

2. Evaluasi Individu

   Tahap kedua dari algoritma genetika adalah evaluasi individu, dimana proses ini akan menghitung nilai *fitness* dari setiap kromosom

yang telah dibangkitkan secara random pada tahap inisialisasi populasi di atas. Dalam masalah optimasi, individu (kromosom) yang bernilai *fitness* yang tinggi yang akan bertahan hidup atau yang akan terpilih dan kromosom yang bernilai rendah akan mati atau tidak terpilih pada tahap selanjutnya.Adapun rumus fungsi fitness yang digunakan dapat dilihat pada Persamaan (10).

$$f_i(x) = \frac{\sum_{a=1}^{u} validitas(a)}{u} \qquad (10)$$

di mana:

u = jumlah data training

$f_i(K)$ = fungsi *fitness* untuk K pada populasi ke-*i*

3. Elitisme

Elitisme adalah suatu prosedur pengopian individu agar individu yang bernilai *fitness* terbaik tidak hilang selama proses evolusi. Suatu individu yang memiliki nilai *fitness* terbaik belum pasti akan selalu terpilih. Hal ini disebabkan karena proses penyeleksian dilakukan secara *random*.

4. Seleksi Orang Tua

Seleksi merupakan proses dalam algoritma genetika untuk memilih kromosom yang tetap bertahan dalam populasi. Kromosom yang terpilih mempunyai kemungkinan untuk dipasangkan dengan kromosom lain atau mengalami proses penyilangan sebanding dengan probabilitas penyilangan yang menghasilkan kromosom anak.

5. Proses Penyilangan (*Crossover*)

Penyilangan merupakan operator dalam algoritma genetika yang bertujuan untuk melahirkan kromosom baru yang mewarisi sifat-sifat induknya sebagaimana proses reproduksi yang terjadi dalam kehidupan alam.

6. Proses Mutasi

Mutasi merupakan operator dalam algoritma genetika yang bertujuan untuk mengubah gen-gen tertentu dalam sebuah kromosom. Proses ini dimodelkan sebagaimana yang terjadi dalam kehidupan alam. Probabilitas mutasi dari suatu gen biasanya sangat kecil, persis seperti kejadian sebenarnya dalam kehidupam alamiah yang memungkinkan terjadinya mutasi genetis tetapi dalam persentase yang sangat kecil.

7. Penggantian Populasi

Untuk pergantian populasi dalam suatu generasi digunakan *general replacement* yaitu pergantian populasi secara keseluruhan. Populasi pada generasi sebelumnya yang merupakan *parents* diganti seluruhnya dengan populasi baru yang merupakan anak atau turunannya (*offspring*). Populasi pada generasi berikutnya adalah kromosom bentukan baru hasil pindah silang dan mutasi serta ditambah kromosom hasil elitisme.

Batubara

Saat ini, batubara merupakan komoditas energi yang semakin menarik. Eksplorasi dan eksploitasi batubara terus meningkat untuk pemenuhan kebutuhan energi masyarakat dunia. Batubaramerupakan istilah yang luas untuk keseluruhan bahan bersifat karbon yang terjadi secara ilmiah.

Kualitas batubara adalah sifat fisika dan kimia dari batubara yang dipengaruhi potensi kegunaannya. Umumnya untuk menentukan kualitas batubara yang diantaranya berupa analisis proksimat dan analisis ultimat. Analisis proksimat digunakan untuk mengetahui karakteristik dan kualitas batubara dalam kaitannya dengan penggunaan batubara tersebut. Analisis proksimat dilakukan untuk menentukan jumlah air (*moisture*), zat terbang (*volatile matter*), karbon padat (*fixed carbon*) dan kadar abu (*ash*), sedangkan analisis ultimat adalah analisa dalam penentuan jumlah unsur Karbon (C), Hidrogen (H), Oksigen (O), Nitrogen (N) dan Sulfur (*Sulphur* atau S).

Beberapa negara memiliki sistem klasifikasi batubara secara spesifik. Klasifikasi digunakan untuk menggolongkan batubara berdasarkan pemanfaatannya. Secara luas, klasifikasi batubara terdiri dari aspek komersial dan aspek ilmiah. Klasifikasi barubara menurut kalori yaitu:

1. *Lignite*
2. *Sub-bituminous*
3. *Bituminous*
4. *Anthracite*

Hasil Penelitian dan Pembahasan

1. Statistika Deskriptif

Dalam melakukan analisis statistika deskriptif dilakukan dengan menampilkan grafik serta mencari rata-rata kandungan tujuh variabel yang ditelitiyaitu TM, M, *Ash*, VM, FC, TS dan GCV untuk masing-masing klasifikasi batubara *lignite*, *sub-bituminous* dan *bituminous*. Adapun nilai rata-rata kandungan batubara dapat dilihat pada Tabel 1.

Tabel 1. Rata-rata Kandungan Batubara

Jenis Batubara	M	TM	VM	FC	*Ash*	TS	GCV
Lignite	18,96	34,78	35,86	31,46	13,77	0,91	35,86
Sub-bituminous	14,69	26,36	40,71	39,43	5,16	0,82	40,71
Bituminous	10,8	16,77	40,45	45,39	3,35	0,67	40,45

2. Normalisasi Data

Dalam melakukan normalisasi data digunakan Persamaan (3), (4) dan (5) untuk 60 data klasifikasi batubara.

3. Melakukan Randomisasi Data

Selanjutnya melakukan randomisasi data yang bertujuan agar setiap data memiliki kesempatan yang sama untuk menjadi data *training* dan *testing*.

4. Membagi Data *Training* dan *Testing*

Setelah melakukan randomisasi data, langkah berikutnya membagi data menjadi data *training* dan *testing* yang nanti akan digunakan dalam menentukan klasifikasi batubara dengan algoritma genetika dan algoritma MK-NN. Dengan proporsi 90:10 diperoleh data *training* sebanyak 54 dan data *testing* sebanyak 6.

5. Penentuan Nilai K Optimal dengan Algoritma Genetika

Penelitian ini menggunakan algoritma genetika sebagai salah satu cara yang digunakan untuk melakukan optimasi nilai K yang dipakai ketika melakukan prediksi dengan menggunakan algoritma MK-NN. Sebelum melakukan tahap awal dalam algoritma genetika perlu dilakukan penentuan parameter yang akan digunakan yaitu:

Popsize	: 8
Pc	: 0,7
Pm	: 0,2
Panjang gen	: 6
Jumlah generasi (iterasi)	: 15

Parameter yang ditentukan merujuk pada parameter yang disarankan oleh De Jong.

a. Inisialisasi Populasi

Proses inisialisasi populasi adalah proses membangkitkan kromosom secara acak sebanyak ukuran populasinya yaitu 8 kromosom. Kromosom merupakan representasi calon solusi atau penyelesaian masalah di mana nilai kromosom akan menjadi nilai K yang digunakan pada algoritma MK-NN. Kromosom dibangkitkan dengan menggunakan bilangan biner 0 dan 1 sebanyak panjang gennya yaitu 6 yang ditampilkan pada Tabel 2.

Tabel 2. Inisialisasi Populasi

Individu	Kromosom	Biner
P1	2	10
P2	44	101100
P3	6	110
P4	20	10100
P5	4	100
P6	20	10100
P7	19	10011
P8	34	100010

b. Membuat Matriks Jarak *Euclidean* antar Data *Training* dan Matriks *Similarity*

Perhitungan jarak *Euclidean* antar data *training* dilakukan untuk menentukan jarak terdekat data berdasarkan nilai K yang diperoleh dari kromosom menggunakan Persamaan (1). Contoh perhitungan untuk data *training* 1 terhadap data *training* 1 dan 2 sebagai berikut:

$$d(x_1,x_1) = \sqrt{\sum_{i=1}^{r}(x_{1i}-x_{1i})^2}$$

$$=0$$

$$d(x_1,x_2) = \sqrt{\sum_{i=1}^{r}(x_{1i}-x_{2i})^2}$$

$$=\sqrt{((-0{,}766)-1{,}493)^2+\cdots+(0{,}744-(-1{,}156))^2}$$

$$=5{,}2273$$

Jarak *Euclidean* dihitung sampai dengan data *training* ke 54. Kemudian dilakukan perhitungan *similarity* berdasarkan Persamaan (2) yang berguna untuk proses perhitungan validitas dalam menentukan nilai *fitness*. Adapun tabel nilai perhitungan jarak *Euclidean* antar data *training* dan matriks *similarity*dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Perhitungan Jarak *Euclidean* antar Data *Training*

Data *Training*	1	2	3	4	⋯	54
1	0	5,2274	1,4087	3,2064	⋯	2,9929
2	5,2274	0	4,8720	7,0020	⋯	3,9188
3	1,4087	4,8720	0	2,9774	⋯	2,4080
4	3,2064	7,0020	2,9774	0	⋯	3,2817
⋮	⋮	⋮	⋮	⋮	⋱	⋮
54	2,9929	3,9188	2,4080	3,2817	⋯	0

Tabel 4. Perhitungan *Similarity* antar Data *Training*

Data *Training*	1	2	3	4	⋯	54
1	1	0	1	1	⋯	0
2	0	1	0	0	⋯	0
3	1	0	1	1	⋯	0
4	1	0	1	1	⋯	0
⋮	⋮	⋮	⋮	⋮	⋱	⋮
54	0	0	0	0	⋯	1

c. Melakukan Perhitungan Validitas dan *Fitness* Populasi

Nilai validitas ini merupakan dasar dari perhitungan nilai *fitness* pada algoritma genetika. Perhitungan validitas dilakukan sebanyak ukuran populasi berdasarkan nilai kromosom. Sebagai contoh dilakukan perhitungan dengan menggunakan kromosom pertama yaitu dengan nilai K=2berdasarkan persamaan (6).

$$validitas(1) = \frac{1}{2}\left[s((x_1),(n_1(x_2)))+s((x_1),(n_2(x_{25})))\right]$$

$$=\frac{1}{2}\times(1+1)$$

$$=1$$

Nilai validitas sebesar 1 merupakan nilai tertinggi yang bisa dicapai untuk masing-masing data *training*dan dapat dikatakan bahwa data *training* ke-1 sangat *valid* atau benar.Untuk lebih jelasnya perhitungan validitas disajikan pada Tabel 5.

Tabel 5. Perhitungan Validitas Data *Training*

Data *Training*	Validitas
1	1
2	1
3	0,5
4	1
5	1
⋮	⋮
54	1

Selanjutnya yaitu menghitung nilai *fitness* yang digunakan sebagai fungsi optimasi dalam penentuan nilai K yang paling optimal. Nilai K optimal akan dicapai dengan syarat berhenti jika nilai *fitness*-nya mencapai 0,9 dan dengan maksimal generasi sebanyak 15. Untuk memperoleh nilai *fitness* dilakukan dengan cara menghitung rata-rata validitas berdasarkan Persamaan (10), sebagai contoh perhitungan fungsi *fitness* populasi pertama dengan K=2 adalah sebagai berikut:

$$f_1(\text{K=2}) = \frac{\sum_{a=1}^{54} validitas(a)}{54} = \frac{(1+1+0{,}5+1+\ldots+1)}{54} = 0{,}8981$$

Perhitungan nilai *fitness* dilakukan sampai jumlah populasi ke-8. Perhitungan nilai *fitness* untuk populasi awal dapat dilihat pada Tabel 6.

Tabel 6. Perhitungan Fungsi *Fitness*

Individu	Kromosom	*Fitness*
P1	2	0,8981
P2	44	0,3737
P3	6	0,8796
P4	20	0,6509
P5	4	0,8889
P6	20	0,6509
P7	19	0,6793
P8	34	0,4701

d. Melakukan Penyilangan (*Crossover*)

Penyilangan merupakan operator dalam algoritma genetika yang bertujuan untuk melahirkan kromosom baru yang mewarisi sifat-sifat induknya sebagaimana proses reproduksi yang terjadi dalam kehidupan alam. Pada penelitian ini telah ditentukan bahwa probabilitas *crossover* (Pc) adalah 0,7. Berdasarkan perhitungan maka *offspring* yang diperoleh adalah sebanyak 6 individu. Proses Penyilangan disajikan pada Tabel 7.

Tabel 7. Hasil Proses Penyilangan

Orang Tua	Penyilangan	Hasil	Nilai
44 dan 44	101100 101100	101100	44
34 dan 34	100010 100010	100010	34
6 dan 19	000110 010011	010110	22
6 dan 20	000110 010100	010110	22
19 dan 20	010011 010100	010111	23
4 dan 20	000100 010100	010100	20

e. Melakukan Mutasi

Mutasi merupakan operator dalam algoritma genetika yang bertujuan untuk mengubah gen-gen tertentu dalam sebuah kromosom. Probabilitas mutasi (Pm) dari suatu gen biasanya sangat kecil, pada penelitian ini Pm yang digunakan adalah 0,2.Dari perhitungan diperoleh bahwa *offspring* yang dihasilkan dari proses mutasi sebanyak 2 individu, kemudian dipilih secara acak individu pada populasi yang akan dilakukan mutasi. Lebih jelasnya disajikan pada Tabel 8.

Tabel 8. Hasil Proses Mutasi

Orang Tua	Mutasi	Hasil	Nilai
19	010011	101100	44
19	010011	101100	44

f. Menghitung *FitnessOffspring* dan Evaluasi Model

Offspring yang telah dihasilkan oleh proses penyilangan dan mutasi selanjutnya dilakukan perhitungan nilai *fitness* seperti pada perhitungan sebelumnya. Untuk *offspring* hasil penyilangan diberi simbol C dan *offspring* hasil mutasi diberi simbol M. Selanjutnya dilakukan evaluasi model dengan menggabungkan individu *offspring* hasil penyilangan dan mutasi dan populasi awal pada satu tabel yang menjadi populasi total dan kemudian dihitung nilai *fitness*-nya.Hasil perhitungan evaluasi model yang telah diurutkan dan nilai *fitness* dapat dilihat pada Tabel 9.

g. Melakukan Proses Seleksi

Hasil evaluasi model yang telah diurutkan berdasarkan *rangking*-nya kemudian dilakukan proses seleksi pada tiap kromosom yang akan menjadi populasi baru pada generasi selanjutnya. Pada tahap seleksi ini akan dipilih populasi sebanyak *popsize* sehingga dari seluruh populasi total maka hanya akan dipilih sebanyak 8 kromosom yang memiliki nilai *fitness* yang paling tinggi. Hasil seleksi elitisme disajikan pada Tabel 10.

Tabel 9. Hasil Evaluasi Model yang Telah Diurutkan

Individu	Kromosom	*Fitness*
P1	2	0,8981
P5	4	0,8889
P3	6	0,8796
P7	19	0,6509
P4	20	0,8889
P6	20	0,6509
C6	20	0,6793
C3	22	0,4701
C4	22	0,3737
C5	23	0,4701
C3	34	0,6077
P8	34	0,6077
P2	44	0,5893
C1	44	0,6509
M1	44	0,3737
M2	44	0,3737

Tabel 10. Hasil Seleksi Elitisme

Individu	Kromosom	*Fitness*
P1	2	0,8981
P5	4	0,3737
P3	6	0,8796
P7	19	0,6509
P4	20	0,8889
P6	20	0,6509
C6	20	0,6793
C3	22	0,4701

h. Penentuan Nilai K Optimal

Penentuan nilai K yang optimal diperoleh dari optimasi yang dilakukan dengan menggunakan algoritma genetika melalui kriteria berhenti yang telah ditentukan yaitu maksimal nilai *fitness* yang mencapai 0,9 dan tercapainya batas generasi yang telah ditentukan yaitu 15. Setelah proses algoritma genetika dihentikan maka akan diperoleh populasi baru yang memiliki nilai *fitness* yang paling tinggi. Kromosom yang akan menjadi nilai K optimal adalah kromosom yang memiliki *ranking* 1 dengan nilai *fitness* terbesar. Dengan proporsi data *training* dan *testing* 90:10 menggunakan algoritma genetika diperoleh nilai K optimal yaitu K=1 dengan nilai *fitness* 0,9444, kemudian dengan proporsi 80:20 diperoleh K optimal yaitu K=1 dengan nilai *fitness* 0,9792 dan dengan proporsi 70:30 diperoleh K optimal yaitu K=1 dengan nilai *fitness* 0,8809.

6. Prediksi Klasifikasi dengan Algoritma MK-NN

Prediksi klasifikasi menggunakan algoritma MK-NN merupakan perkembangan dari K-NN tradisional yang bertujuan untuk meningkatkan akurasi dari K-NN, dengan menambahkan fungsi validitas dan *Weight Voting*.

a. Menghitung Jarak *Euclidean* Data *Training* dengan Data *Testing*

Dengan menggunakan proporsi 90:10 dilakukan perhitungan jarak *Euclidean* data *training* dan data *testing* dengan menggunakan Persamaan (1). Adapun perhitungan jarak *Euclidean*data *training* 1 dan data *testing* 1 adalah sebagai berikut:

$$d(x_1,y_1) = \sqrt{\sum_{i=1}^{r}(x_{1i}-y_{1i})^2}$$
$$= \sqrt{((-0{,}766)-0{,}371)^2+\cdots+(0{,}744-(-1{,}565))^2}$$
$$= 5{,}0216$$

Perhitungan jarak *Euclidean* dilakukan pada data *testing* 1 sampai dengan data *testing* 6 dan diulang sebanyak 54 data *training* agar memperoleh tetangga terdekat terhadap data *testing.* Untuk hasil perhitungan jarak *Euclidean* data *training* dan data *testing* disajikan pada Tabel 11.

Tabel 11. Perhitungan Jarak *Euclidean* Data *Training* dan Data *Testing*

Data *Training*	Data *Testing*					
	1	2	3	4	5	6
1	5,0216	6,7064	2,8594	3,3243	2,9257	2,6131
2	2,7335	5,7302	4,6904	3,1922	5,3117	3,2200
3	5,0331	7,1163	2,6936	2,9573	2,7786	2,3762
4	6,7706	7,8764	2,8828	4,0748	2,1918	4,4878
⋮	⋮	⋮	⋮	⋮	⋮	⋮
53	3,3519	5,7395	2,2172	1,2805	2,6642	1,3107
54	4,3662	6,3383	1,3085	1,1497	1,8476	2,4673

b. Menghitung Nilai *Weight Voting*

Perhitungan *weight voting* dilakukan untuk mencari bobot data *testing* terhadap data *training*-nya. Dengan menggunakan K=1 maka dicari tetangga terdekat untuk masing-masing data *testing*. Adapun jarak *Euclidean* terdekat masing-masing data *testing* disajikan pada Tabel 12.

Setelah mengetahui jarak *Euclidean* terhadap masing-masing data *testing* dilakukan perhitungan *weight voting* pada data *testing* 1 sebagai berikut:

$$w(x_{48},y_1) = validitas(48)\times\frac{1}{d(x_{48},y_1)+0{,}5}$$

$$=1\times\frac{1}{1,3490+0,5}$$

$$=0,5408$$

Perhitungan diulangi sampai data *testing* ke enam dan berikutnya dilakukan *voting* kelas pada data *testing*.

Tabel 12. Jarak *Euclidean* Terdekat Data *Testing*

Data *Testing*	Jarak Terdekat Pada Data *Training*	Kelas Data *Training*
1	48	1
2	19	1
3	35	3
4	34	2
5	18	3
6	53	2

c. Melakukan *Voting* pada Data *Testing* untuk Menentukan Kelas Prediksi

Nilai *weight voting* yang telah diperoleh dari data *testing* selanjutnya dilakukan *voting* berdasarkan besarnya *weight voting* tergantung pada kelas data *training*-nya. Karena K optimal K=1 maka hanya terdapat satu nilai *weight voting* untuk masing-masing kelas dan kelas yang memiliki nilai *voting* tertinggi akan menjadi kelas data *testing*. Sebagai contoh perhitungan voting pada data *testing* 1 sebagai berikut:

$$vote(Kelas) = \sum_{i=1}^{n} w(x_a, y_b)_n$$

$$vote(Kelas(1)) = \sum_{i=1}^{1} w(x_a, y_b)_1$$

$$=w(x_{48}, y_1)$$

$$=0,5408$$

$$vote(Kelas(2)) =0$$

$$vote(Kelas(3)) =0$$

Voting dilakukan sampai data *testing* ke enam.

7. Menghitung Prediksi Akurasi Klasifikasi Algoritma MK-NN

Setelah dilakukan perhitungan *voting* pada data *testing* untuk menentukan kelas prediksi, dilakukan perhitungan akurasi klasifikasi dengan menggunakan K=1 pada 6 data *testing*. Berdasarkan perhitungan *voting* diperoleh perbandingan kelas prediksi data *testing* dan kelas data *testing* aslinya yang disajikan pada Tabel 13. Berdasarkan Tabel 13 dapat dilihat bahwa ke enam prediksi kelas data *testing* memiliki kelas yang sama dengan kelas data aslinya. Proses selanjutnya yaitu menghitung prediksi akurasi klasifikasi MK-NN dengan K=1 dengan menggunakan Persamaan (9) sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah data yang diprediksi benar}}{\text{jumlah prediksi yang dilakukan}}\times 100\%$$

$$= 100\%$$

Tabel 13. Hasil Perbandingan Kelas Prediksi Data *Testing* Dan Kelas Aslinya

Data Testing	Kelas Prediksi	Kelas Data Asli	Keterangan
1	1	1	Benar
2	1	1	Benar
3	3	3	Benar
4	2	2	Benar
5	3	3	Benar
6	2	2	Benar

Berdasarkan perhitungan prediksi akurasi klasifikasi proporsi data *training* dandata *testing* 90:10 diperoleh bahwa dengan K=1 memiliki akurasi klasifikasi sebesar 100%. Setelah dilakukan proses yang sama dengan proporsi data *training* dandata *testing* 80:20 dan 70:30 diperoleh juga nilai K optimal yang sama yaitu K=1.Selanjutnya dilakukan perhitungan prediksi akurasi melalui proses yang sama dengan menggunakan proporsi data 80:20 diperoleh akurasi klasifikasi sebesar 91,67% dan dengan menggunakan proporsi data 70:30 diperoleh akurasi klasifikasi sebesar 94,44%.

Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Nilai K optimal yang diperoleh dengan menggunakan algoritma *Genetic Modified* K-*Nearest Neighbor* (GMK-NN) untuk memprediksi klasifikasi batubara di PT Jasa Mutu Mineral Indonesia pada penggunaan proporsi data *training* dan data *testing* sebesar 90:10, 80:20 dan 70:30 adalah K=1.
2. Persentase akurasi prediksi klasifikasi batubara di PT Jasa Mutu Mineral Indonesia dengan menggunakan algoritma *Genetic Modified* K-*Nearest Neighbor* (GMK-NN) pada proporsi data *training* dan data *testing* sebesar 90:10 adalah 100%, lalu pada proporsi 80:20 adalah 91,67% dan pada proporsi 70:30 adalah 94,44%.

Daftar Pustaka

Arif, Irwandy. (2014). *Batubara Indonesia*. Jakarta: Gramedia Pustaka Utama.

Mutrofin, Siti. (2015). Optimasi Teknik Klasifikasi *Modified* K-NN Menggunakan Algoritma Genetika. *Jurnal GAMMA ISSN 0216-9037*.

Parvin, H., Alizadeh, H., dan Bidgoli, B.M. (2010). A Modification on K-Nearest

Neighbor Classifier. *Global Journal of Computer Science and Technology Vol. 10 November 2010*.

Prasetyo, Eko. (2012). *Data Mining: Konsep Dan Aplikasi Menggunakan Matlab*. Yogyakarta: ANDI.

Prasetyo, Eko. (2014). *Data Mining: Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: ANDI.

Rodiyansyah, S., dan Winarko, Edi. (2013). Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayes Bayesian Classification. *Jurnal Universitas Pendidikan Indonesia*.

Sukandarrumidi. (2017). *Batubara dan Pemanfaatannya: Pengantar Teknologi Batubara Menuju Lingkungan Bersih*. Yogyakarta: UGM Press.

Zukhri, Zainudin. (2014). *Algoritma Genetika: Metode Komputasi Untuk Menyelesaikan Masalah Optimasi.* Yogyakarta: ANDI.