# Optimasi Parameter *K* Pada Algoritma K-NN Untuk Klasifikasi Prioritas Bantuan Pembangunan Desa

*Optimization of K Parameters in the K-NN Algorithm for Priority Classification of Village Development Assistance*

Saiful Ulya[1], M. Arief Soeleman[2], Fikri Budiman[3]
[1,2,3]Magister Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
E-mail: [1]ipul.ulya@gmail.com, [2] m.arief.soeleman@dsn.dinus.ac.id,
[3]fikri.budiman@dsn.dinus.ac.id

**Abstrak**

Klasifikasi adalah proses menemukan model atau fungsi yang menggambarkan dan membedakan kelas atau konsep data. Algoritma k-NN *(k Nearest Neighbors)* merupakan algoritma klasifikasi berdasarkan pembelajaran dari data yang sudah terklasifiasi sebelumnya. Algoritma k-NN *(k Nearest Neighbors)* merupakan algoritma yang sangat bagus dalam menangani beberapa kasus, salah satu kelebihan k-NN diantaranya adalah tangguh terhadap *data training* yang noisy dan sangat efektif apabila data trainingnya besar. Namun terdapat beberapa masalah pada algoritma k-NN diantaranya adalah penentuan nilai *k* untuk pemilihan jumlah tetangga terdekatnya sangat sulit, karena nilai *k* sangat peka atau sensitif terhadap hasil klasifikasi. Pada penelitian ini, akan dilakukan pemodelan klasifiasi dengan menggunakan algoritma k-NN yang difokuskan pada proses penentuan nilai *k* terbaik pada dataset IKG (Indeks Kesulitas Geografis) desa. Pada penelitian ini akan melakukan integrasi algoritma k-NN dengan menentukan nilai *k* optimal dengan *optimize parameters* berdasar algoritma genetika.

Kata Kunci*: Data Mining,* Klasifikasi, *k-Nearest Neighbors,* Algoritma Genetika, *Optimize Parameter.*

***Abstract***

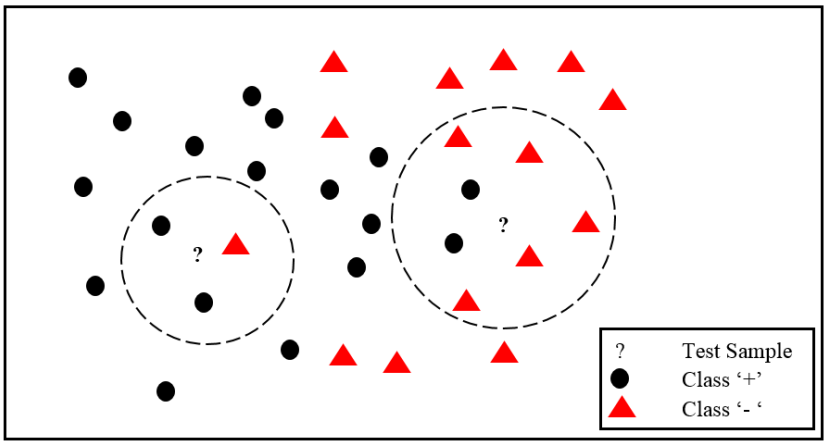
*Classification is the process of finding a model or function that describes and distinguishes classes or concepts of data. The k-NN (k-Nearest Neighbors) algorithm is a classification algorithm based on learning from previously classified data. The k-NN algorithm (k-Nearest Neighbors) is a very good algorithm in handling several cases, one of the advantages of k-NN is that it is very tough on noisy training data and is very effective when the training data is large. However, there are some problems in the k-NN algorithm including the determination of the value of k for the selection of the number of closest neighbors is very difficult, because the value of k is very sensitive or sensitive to the results of classification. In this study, classification modelling will be carried out using the k-NN algorithm which is focused on the process of determining the best k value in the IKG dataset (Geographic Difficulty Index). This research will integrate the k-NN algorithm by determining the optimal k value by optimizing parameters based on genetic algorithms.*

*Keywords: Data Mining, Classification, k-Nearest Neighbors, Genetic Algorithms, Optimize Parameters.*

## 1. PENDAHULUAN

Klasifikasi adalah proses menemukan model atau fungsi yang menggambarkan dan membedakan kelas atau konsep data [1]. Model diturunkan berdasarkan analisis satu dataset training (yaitu, objek data yang label kelasnya sudah diketahui). Kemudian model ini digunakan untuk memprediksi label kelas objek data testing yang label kelasnya tidak atau belum diketahui. Klasifikasi juga merupakan metode yang populer karena mudah dijelaskan, sangat selaras dengan apa yang diasosiasikan sebagian besar orang pada model "terbaik", dan ini mengukur model yang cocok untuk semua nilai [2]. Metode atau algoritma yang sering digunakan untuk klasifikasi diantaranya *Naïve Bayes, Decission Tree, Neural Network, Support Vector Machine, Logistic Regresion, Linear Regresion,* dan k-NN *(k Nearest Neighbors)* [1].

Algoritma k-NN *(k Nearest Neighbors)* merupakan algoritma klasifikasi berdasarkan pembelajaran dari data yang sudah terklasifiasi sebelumnya. Algoritma ini termasuk *supervised learning* dimana hasil dari *instance* yang baru diklasifikasikan berdasar mayoritas jarak tetangga terdekat dari kelas yang ada. Sedangkan menurut *Zhang* menjelaskan bahwa algoritma k-NN *(k Nearest Neighbors)* merupakan metode *non parametrik* atau berbasis *instance* dan telah dianggap sebagai salah satu metode sederhana dalam *data mining* dan *machine learning* [3]. Prinsip algoritma k-NN adalah bahwa sampel yang paling mirip dengan kelas yang sama dari data training memiliki probabilitas paling tinggi. Algoritma k-NN sangat bergantung pada penentuan nilai *k* untuk memilih jumlah tetangga terdekatnya berdasar urutan dari nilai yang paling kecil dari hasil perhitungan jarak atau *distance*. Proses klasifikasi ditentukan dengan memilih suara terbanyak dari tetangga. Pada Gambar 1.1 menjelaskan tentang bagaimana proses klasifikasi dengan k-NN berdasarkan nilai k=3 pada bagian kiri dan nilai k=7 pada bagian kanan.




Gambar 1. Contoh pelatihan pada klasifikasi k-NN [4]

Dari gambar diatas dijelaskan bahwa ketika menentukan nilai k=3 untuk algoritma k-NN, ada dua kelas '+' yang diprediksi dari sampel uji sesuai aturan algoritma k-NN. Kemudian saat menentukan nilai k=7 terdapat 5 kelas '-' dari data sampel uji sesuai aturan algoritma k-NN dimana nilai tersebut merupakan jumlah suara terbanyak dari tetangga terdekat.

Algoritma k-NN *(k Nearest Neighbors)* merupakan algoritma yang sangat bagus dalam menangani beberapa kasus, terbukti k-NN *(k Nearest Neighbors)* termasuk dalam sepuluh algoritma terbaik dalam *data mining* dan *machine learning* [4] [5] [6] [7], [8]. Salah satu kelebihan k-NN diantaranya adalah tangguh terhadap *data training* yang noisy dan sangat efektif apabila data trainingnya besar. Seperti diketahui, algoritma k-NN sangat peka terhadap penentuan nilai *k* karena nilai *k* tersebut akan sangat menentukan hasil klasifikasi. Meskipun banyak upaya penelitian telah difokuskan pada masalah ini, namun menentukan nilai *k* pada algoritma k-NN masih sangat sulit [9]. Pada beberapa kasus penentuan nilai *k* pada algoritma k-NN para peneliti

masih menggunakan cara tradisional yaitu dengan menentukan *range* kemudian di pilih dengan cara *trial and error*. Namun menurut S. Zhang menyebutkan bahwa cara tradisional tersebut telah terbukti tidak praktis dalam menangani beberapa kasus [5]. Menurut Lall dan Sharma nilai *k* harus memenuhi syarat yaitu $k < N$ dimana N merupakan jumlah record *dataset training* [10]. Nilai k digunakan untuk mencari jumlah mayoritas dari target atau label pada data traning, sehingga nilai *k* tidak boleh lebih dari jumlah record pada *dataset training*. Sebagai dampak dari masalah tersebut, maka penentuan nilai *k* pada algoritma k-NN telah menjadi topik penlitain yang sangat menarik dalam *data mining* dan *machine learning.*

Indeks desa membangun adalah parameter yang digunakan untuk melihat perkembangan suatu desa berdasar beberapa indikator yaitu Indeks Ketahanan Ekonomi (IKE), Indeks Ketahanan Sosisal (IKS), dan Indeks Ketahanan Lingkungan (IKL) [11]. Ada lima kategori desa yang kemudian dapat ditentukan dari tiga indeks tersebut yaitu kategori sangat tertinggal, tertinggal, berkembang, maju, dan mandiri [12].

Dalam sebuah penelitian yang dilakukan oleh CV. Tirta Utama, untuk menentukan besaran nominal anggaran dana desa yang didapat suatu desa yang berdasar pada PP Nomor 60 Tahun 2014 tentang dana desa. Tiga komponen penyusunan IKG adalah ketersediaan pelayanan dasar, kondisi infrastuktur, dan akses trasnportasi. Dari ketiga komponen tersebut ditentukan sebanyak 16 parameter unutk menyusun IKG, parameter IKG ditampilkan pada Tabel 1.

Tabel 1. Parameter Dataset IKG

| No | Nama Atribut |
|---|---|
| 1 | Jumlah Penduduk |
| 2 | Kemiskinan (Orang) |
| 3 | Jumlah SMP/MTs Negeri |
| 4 | Jumlah SMP/MTs Swasta |
| 5 | Jarak SMP/MTs (KM) |
| 6 | Jumlah Rumah Sakit |
| 7 | Jarak Rumah Sakit (KM) |
| 8 | Jumlah Puskesmas |
| 9 | Jarak Puskesmas (KM) |
| 10 | Jumlah Pusesmas Pembantu |
| 11 | Jarak Puskesmas Pembantu (KM) |
| 12 | Faskes Terdekat (KM) |
| 13 | Kerusakan Jalan (%) |
| 14 | Ketinggian Desa Dari Permukaan Laut (M) |
| 15 | Luas Wilayan Desa (KM) |
| 16 | Jarak Desa ke Kabupaten (KM) |

Pada penelitian ini, akan dilakukan suatu pemodelan baru dengan menggunakan dataset yang belum digunakan pada penelitian sebelumnya dengan menggunakan algoritma k-NN yang difokuskan pada proses penentuan nilai *k* terbaik pada dataset IKG (Indeks Kesulitas Geografis) desa. Model Klasifikasi untuk menentukan prioritas bantuan yang telah ada dan dilakukan sebelunya dirangkum dalam Tabel 2.

Tabel 2. Model Klasifiksi Penerimaan Bantuan Sebelumnya

| No | Judul, Penulis | Metode | Keseimpulan |
|---|---|---|---|
| 1 | Klasifikasi Penentuan Penerima Manfaat Program Keluarga Harapan (Pkh) Menggunakan Algoritma C5.0 (Studi Kasus: Desa Sukamaju, Kec.Kadudampit) Dede Wintana, Hikmatulloh, Nurul Ichsan, Jajang Jaya Purnama, Ami Rahmawati. | Klasifikasi dengan Algoritma C5.0 | Algoritma C5.0 dapat digunakan sebagai metode klasifikasi dalam penunjang keputusan penerima manfaat program keluarga harapan dengan memperhatikan nilai gain (penguatan) tertinggi dari empat atribut |
| 2 | Penerapan Metode Naïve Bayes Dalam Klasifikasi Kelayakan Keluarga Penerima Beras Rastra (Chairul Fadlan1, Selfia Ningsih2, Agus Perdana Windarto3) | Klasifikasi dengan Naïve Bayes | Hasil klasifikasi menggunakan Naïve Bayes cukup baik, namun jumlah record dataset yang digunakan terlalu sedikit, sehingga perlu dilakukan pembahan jumlah record |
| 3 | Klasifikasi Penerima Dana Bantuan Desa Menggunakan Metode Knn (K-Nearest Neighbor) (Riyan Latifahul Hasanah1; Muhamad Hasan2; Witriana Endah Pangesti3; Fanny Fatma Wati4; Windu Gata5) | Klasifikasi dengan Algoritma k-Nearest Neighbor | Dengan menggunakan k-NN dengan metode penentuan nilai k secara manual diperoleh hasil akurasi klasifikasi sebesar 81,25%. Model ini cuku baik, namun penentuan nilai k masih menggunakan cara yang belum efisien. |

Seperti diketahui, algoritma k-NN sangat sensitif terhadap penentuan nilai *k* karena nilai *k* tersebut akan mempunyai *impact* atau pengaruh yang sangat kuat terhadap hasil klasifikasi [6]. Untuk mencari nilai optimal beberapa peneliti banyak menggunakan algoritma metaheuristik seperti algoritma genetika [13] [14]. Ada beberapa kelebihan yang dimiliki algoritma genetika daripada algoritma optimasi tradisional yang lain, dua diantara kelebihan tersebut yaitu dapat dan mampu menangani masalah yang kompleks dan pararel. Algoritma genetika dapat menyelesaikan berbagai macam optimasi tergantung pada *object function (fitness)* apakah seimbang atau tidak seimbang, *linier* atau tidak *linier*, berkesinambungan atau tak berkelanjutan, atau dengan *random noise* [15]. Selain dapat menentukan nilai k yang optimal, gabungan antara algoritma genetika dan k-NN juga mampu mengurangi kompleksitas dari algoritma k-NN, dikarenakan penghitungan bobot data training sudah tidak dipertimbangkan lagi. Selain kompeksitas, gabungan antara algoritma genetikan dan k-NN juga dapat dan mampu meningkatkan nilai akurasi dari k-NN [16].

Pada penelitian ini akan menggunakan algoritma k-NN dengan menentukan nilai *k* optimal dengan *optimize parameters* berbasis algoritma genetika, sehingga diharapkan mampu memodelkan hasil klasifikasi yang efisien pada dataset IKG (indeks kesulitan geografis) dengan algoritma k-NN *(k Nearest Neighbor)*.

## 2. METODE PENELITIAN

### *2.1 Pengumpulan Data*

Data ini diperoleh dari hasil survey sebuah lembaga penelitian, dataset ini bersifat privat (tidak terdapat di *library dataset* umum). Dataset ini sebelumnya dijadikan sebagai penelitian untuk menyesuaikan penerimaan Anggaran Dana Desa (ADD) untuk masing-masing desa berdasarkan indeks kesulitas geografis desa.

Dari 227 (dua ratus dua puluh tujuh) *record* di dataset, terdapat dua jenis kelas yaitu label sudah dan label belum. Dua kategori tersebut akan dijadikan pedoman untuk memberikan bantuan pembangunan masing-masing desa dengan nominal yang telah ditentukan oleh dinas terkait

sesuai dengan kategori masing-masing desa

### *2.2 Corelation Matrix*

*Correlation Matrix* adalah proses analisa dari dataset yang digunakan untuk memperoleh informasi keterkaitan antar masing-masing atribut. Hasil analisa korelasi dapat digunakan untuk menjawab pertanyaan diantaranya atribut apa saja yang paling berhubungan untuk menentukan daerah tersebut termasuk kategori prioritas pembangunan. Proses analisis korelasi harus membentuk sebuah matriks korelasi terlebih dahulu. Matriks korelasi berisi nilai koefisien korelasi. Nilai ini mempunyai rentan jarak antara -1 sampai 1. Kemudian nilai ini yang akan menunjukkan keterkaitan dari masing-masing atribut. Nilai positif (0 s/d 1) memberi informasi bahwa atribut saling berbanding lurus, sedangkan nilai negative (-1 s/d 0) menunjukkan bahwa atribut saling berbanding terbalik [17].

Formula untuk mengitung korelasi antar data sampel adalah sebagai berikut.

$$kor\ (x,y) = \sqrt{xy} = \frac{\sum_{i=1}^{n}(xi-\bar{x})(yi-\bar{y})}{\sqrt{\sum_{i=1}^{n}(xi-\bar{x})^2\ \sum_{i=1}^{n}(yi-\bar{y})^2}} \quad (4)$$

Dimana:

$\sum$ = Menghitung penjumlahan

$(xi - \bar{x})$ = nilai x dikurangi nilai rata-ratanya, $\bar{x}$

$(yi - \bar{y})$ = nilai y dikurangi rata-ratanya, $\bar{y}$

### *2.3 Metode k-NN*

Algoritma k-NN *(k Nearest Neighbour)* biasa dikenal juga sebagai "Lazy Learner" [18]. Menurutut Souza, Rittner, & Lotufo pada (Harafani, 2018) algoritma k-NN juga merupakan perpanjangan dari algoritma pengklasifikasi nearest neighbour (NN clasifier) [19]. Kerangka algoritma k-NN pertama kali diperkenalkan oleh *Fix & Hodges* pada tahun 1951 [20]. Meskipun k-NN resmi diajukan lebih dari 60 tahun yang lalu, namun k-NN masih sangat populer sampai dengan saat ini, karena banyak sekali penelitian yang menggunakan metode ini. Sebagai contoh *Bhuvaneswari & Therese* melakukan penelitian untuk mendeteksi penyakit kanker paru-paru [21], selanjutnya *Wakahara & Yamashita* melakukan penelitian klasifikasi karakter tulisan tangan [22], kemudian *Zhang, Deng, Cheng, & Zhu* melakukan klasifikasi pada data besar [23], dan masih terdapat banyak penelitian lain dengan menggunakan k-NN. Metode ini adalah membandingkan kesamaan data pelatihan yang paling dekat dengannya. Nilai *k* mewakili tetangga yang dibandingkan. Pseucode algoritma klasifikasi k-NN ditunjukkan pada Gambar 3.

```
k-Nearest Neighbor
Classify (X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for i = 1 to m do
  Compute distance d(X_i, x)
end for
Compute set I containing indices for the k smallest distances d(X_i, x).
return majority label for {Y_i where i ∈ I}
```

Gambar 2. Pseucode k-NN klasifiasi

Kesamaan data testing dan data training dihitung berdasarkan jarak, secara umum menggunakan penghitungan *euclidian distance* (jarak euclidean). Jarak dari dua nilai, misalnya X1 dan X2, dimana X1= (x11, x12, x13, ....x1n) dan X2= (x21, x22, x23, ....x2n) didefinisikan oleh:

$$dist(qi, pi) = \sqrt{\sum_{i=1}^{n}(qi - pi)^2} \qquad (1)$$

Normalisasi min-max merupakan cara yang digunakan untuk mengubah nilai v dari atribut numerik A ke $v^1$ di kisaran 0 ini didefinisikan oleh:

$$v^1 = \frac{v - Min_A}{Max_A - Max_A} \qquad (2)$$

Didalam *pattern recognition* (pengenalan pola), algoritma k-NN *(k Nearest Neighbor)* dikenal sebagai metode non-parametrik [21] yang digunakan untuk regresi dan klasifikasi. Dalam algoritma K-NN klasifikasi, output merupakan keanggotaan kelas. Klasifikasi dilakukan dengan mengambil suara terbanyak dari tetangga yang yang ditentukan. Jika K = 2.

### *2.4 Algoritma Genetika*

Algoritma genetika (GA), yang dikembangkan oleh John Holland pada dasarnya membentuk dasar komputasi evolusioner modern dan menjadi algoritma evolusioner terpopuler [24], prinsip dasar yang dipergunakan pada algoritma ini adalah dengan menggunakan prinsip dasar seleksi alam yang dikenalkan oleh Charles Darwin. Algoritma genetika seringkali digunakan sebagai pendekatan untuk mengidentifikasi pencarian nilai dan solusi berbagai permasalahan optimasi [25]. Tiga operator genetik utama pda algoritma genetika yaitu: *crossover* (proses penggantian kromosom), *mutasi* (proses penggantian satu solusi untuk menambahkan jenis populasi), dan seleksi (penggunaan solusi dengan nilai fitness yang tinggi untuk lulus ke generasi berikutnya).

Menurut *Haupt* dalam *(Randy & Sue, 2004)* diperlukan langkah-langkah yang perlu dilakukan untuk menyelesaikan masalah-masalah dalam optimasi [26]:

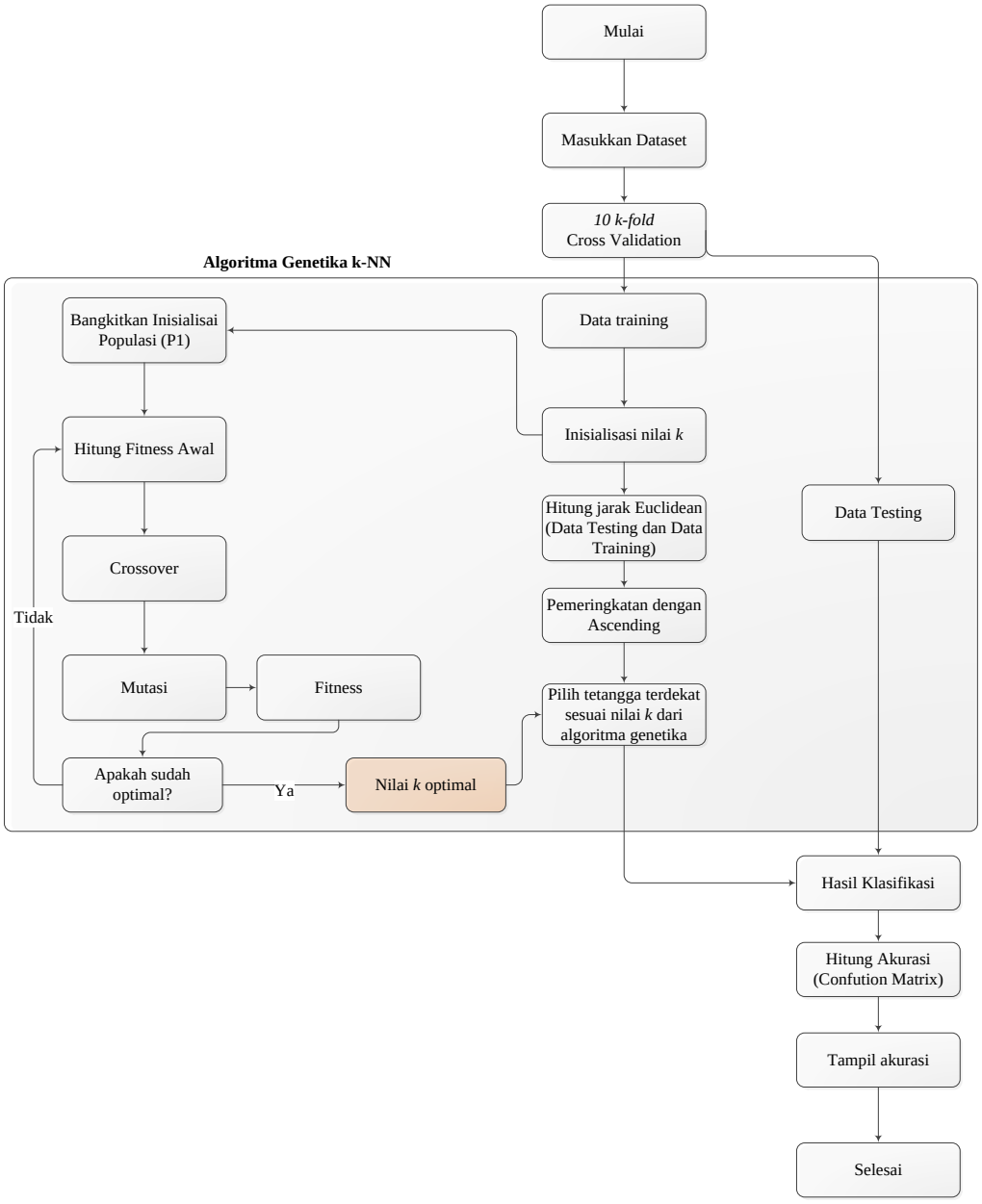
1. Inisialisasi populasi
2. Evaluasi populasi
3. Seleksi populasi
4. Proses penyilangan kromosom *(crossover)*
5. Evaluasi populasi baru
6. Selama syarat belum terpenuhi diulang dari proses 3

Algoritma genetika juga mampu mengatasi jenis-jenis optimasi tergantung pada fungsi objektifnya (fitness) apakah seimbang ataukan tidak seimbang, berkesinambungan atau tidak berkesinambungan, linier atau tidak linier atau dengan *random noise*. Adapun fungsi fitness secara general yang ada pada *Kuceva* pada tahun 1997 [15] dapat dilihat pada Persamaan 1 dibawah ini.

$$J(S) = \sum_{j=1}^{n} hs(Zj) \qquad (3)$$

dimana *hs(Zj)* menyatakan jumlah yang dikontribusikan *Zj* untuk penilaian keseluruhan kriteria, diberikan S, dan menggunakan aturan ketentuan klasifikasi algoritma k-NN.

Model yang diusulkan untuk penelitian ini digambarkan pada Gambar 4.




Gambar 3. Kerangka Model Yang Diusulkan

Penjelasan dari Gambar 2 adalah sebagai berikut

1. Imput data training. Dalam penelitian ini *data training* yang digunakan adalah dataset indeks kesulitan geografis sebanyak 227 *record.*
2. Tentukan populasi dari kromosom (solusi). Misal populasi yang di kehendaki adalah 10 (sepuluh). Maka secara random akan dibangkitkan kromosom (kemungkinan solusi) sebanyak 10 buah, dengan ketentuan, nilai k < jumlah *data training* (pengujian ini menggunakan 10 *k-fold cross validation* yaitu 204 sebagai *data training* dan 23 sebagai *data testing*). Misal hasil kromosom didapatkan 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Selanjutnya nilai kromosom tersebut dibinerkan, misal menjadi 0001 untuk kromosom 1, 0010 untuk kromosom 2, 0011 untuk kromosom 3, 0100 untuk kromosom 4, 0101 untuk kromosom

5, 0110 untuk kromosom 6, 0111 untuk kromosom 7, 1000 untuk kromosom 8, 1001 untuk kromosom 9, dan 1010 untuk kromosom 10.

3. Hitunglah nilai fitnes dengan menggunakan nilai validitas (*validity*), nilai *validity* tertinggi adalah nilai fitness terbaik. Misal yang terbaik adalah kromosom 6.

   $$S\ (a.b) = \begin{cases} 1\ jika\ a = b \\ 0\ jika\ a\ \neq b \end{cases}$$

   Keterangan:
   *S* : nilai similaritas antar data latih
   *a dan b* : label kelas antar data latih

   $$Validity\ (x) = \frac{1}{h}\sum_{i-1}^{h} S(lbl(x), lbl(N_i(x)))$$

   Keterangan:
   *i* : banyaknya data latih
   *k* : jumlah teteangga terdekat antar data latih dari similaritas yang terbaik (nilai 1=terbaik), $N_i$ adalah banyaknya label kelas
4. Lakukan proses seleksi menggunakan *roulette wheel*, misalkan yang terpilih adalah kromosom 6 dan 9.
5. Lakukan *crossover* dari kedua kromosom tersebut (langkah 4), dikarenakan operasi ini diharapkan memiliki keberhasilan tinggi, maka digunakan probabilitas sebesar 0.8.
6. Lakukan mutasi dari hasil anakan pada langkah no. 5, dikarenakan operasi ini diharapkan memiliki kemungkinan kecil, maka digunakan probabilitas sekecil-kecilnya, misal 0.001.
7. Diperoleh individu baru dari nilai fitness yang terbaik.
8. Ulangi tahapan tahapan di operasi GA sampai menemukan kromosom yang optimal kemudian diguakan sebagai nilai *k*.
9. Lakukan perhitungan *Euclidean distance* dari data uji ke data latih.\

   $$dist(qi, pi) = \sqrt{\sum_{i=1}^{n}(qi - pi)^2}$$

   Keterangan:
   *dist (qi,pi)* : Jarak antara data uji dan data latih
   *x* : Data uji
   *y* : Data latih
   *n* : Jumlah data latih
10. Lakukan perhitungan bobot (w) dengan mempertimbangan nilai validitas dan jarak.

    $$w(i) = validitas(i)x\frac{1}{d_e + a}$$

    Keterangan:
    *w* : bobot
    *a* : nilai smooting (pemulusan), dalam penelitian nilai yang digunakan adalah 0,5
11. Nilai bobot terbesar merupakan prediksi kelas dari data uji.

## 3. HASIL DAN PEMBAHASAN

Bagian ini akan melakukan penentuan nilai *k* pada aloritma k-NN menggunakan metode yang diusulkan yaitu optimasi parameter berbasis Algoritma Genetika. Pengujian pertama ini akan dilakukan sebanyak 10 kali pengujian.

Selanjutnya pada bagian ini, pengujian dimulain dengan memasukkan dataset kemudia masuk pada proses *cross validation*, dibagian *cross validation* menggunakan metode *10 k-fold* atau dataset dilipat menjadi 10 bagian dengan 1 lipatan sebagai *data testing* dan 9 lipatan lainnya sebagai *data training* dan dilakukan beulang-ulang (10 kali pengujian) sampai masing-masing lipatan merasakan sebagai *data testing.*

Setelah itu data trainig masuk ke proses k-NN yang kemudian saat menginisiasi nilai *k*

proses di inisiasi dimasukkan kedalam algoritma genetika untuk membangkitkan populasi (p1), kemudian pada proses k-NN lanjut ke tahapan hitung jarak dengan *Euclidean distance* dimana jarak antara *data training* disetiap kromosm disetiap *data testing*, sebagai nilai *fitness*. Kemudian pilihlah *kromosom* dengan nilai *fitness* tertinggi sebagai *Global Maximum (GMax).* Selanjutnya lakukan reproduksi (seleksi) kemudian terapkan proses *crossover* dan terapkan operator mutase untuk mendapatkan individu baru (p2). Selanjutnya hitung *local maximum (LMax).* Jika *GMax < LMax* maka *Gmax=Lmax;* atau p1=p2; selanjutnya kromosom yang memiliki *Gmax* dijadikan sebagai nilai *k* yang optimal kemudian digunakan untuk memilih jumlah tetangga terdekat berdasar pemeringkatan *ascending* dari hasil perhitungan *Euclidean distance* yang telah diproses pada k-NN, setelah itu akan tampil hasil klasifikasi dari *10 k-fold cross validation* kemudian hitung akurasi menggunakan *confusion matrix* dan akan tampil hasil akurasi dari pengujian.

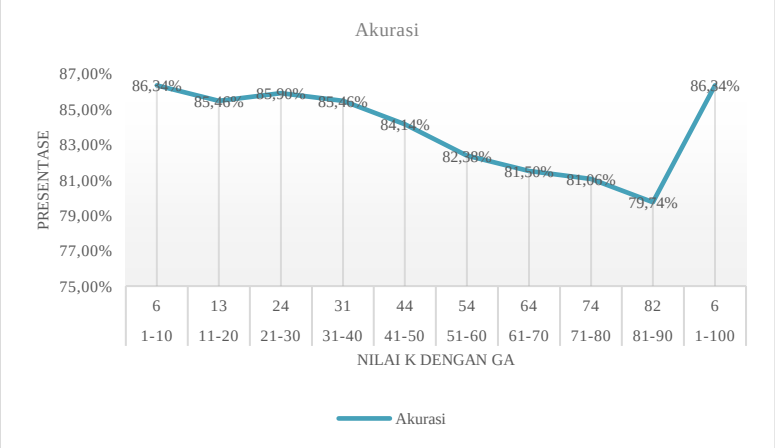
Hasil rangkuman masing-masing pengujian disajikan dalam bentuk tabel yang dapat dilihat pada Tabel 4.

Tabel 3. Hasil Pengujian dengan Optimasi Parameter Algoritma Genetika

| Range | Nilai K | Akurasi | Recall | Presision |
|---|---|---|---|---|
| 1-10 | 6 | 86,34% | 84,76% | 95,86% |
| 11-20 | 13 | 85,46% | 83,33% | 96,55% |
| 21-30 | 24 | 85,90% | 83,83% | 96,55% |
| 31-40 | 31 | 85,46% | 83,73% | 95,86% |
| 41-50 | 44 | 84,14% | 82,25% | 95,86% |
| 51-60 | 54 | 82,38% | 81.44% | 93.79% |
| 61-70 | 64 | 81,50% | 80,47% | 93,79% |
| 71-80 | 74 | 81,06% | 78,98% | 95,86% |
| 81-90 | 82 | 79,74% | 77,97% | 95,17% |
| 1-100 | 6 | 86,34% | 84,76% | 95,86% |

Dari Tabel 4. dapat dijelaskan bahwa nilai *k* yang ada pada kolom nomor 2 merupakan hasil nilai optimal (fitness) dari masing-masing range. Jika dilihat dari hasil pengujian sebelumnya pada pengujian ini didapatkan hasil yang lebih baik bahwa nilai k optimal diperoleh dari hasil fitness range 1-10 dan range 1-100 dengan nilai *k=6* dan dengan akurasi 86,34%.

Berikut grafik hasil pengujian dengan optimasi parameter berbasis algoritma genetika dapat dilihat pada Gambar 5.




Gambar 4. Grafik Pengujian dengan Optimasi Parameter

Dari apa yang ditampilkan pada Gambar 5. diatas dapat disimpulkan bahwa dari pegujian algoritma k-NN dengan menggunakan algoritma genetika untuk menentukan nilai *k* terbaik untuk dataset klasifikasi Indeks Kesulitan Geografis tersebut diperoleh nilai *k* optimal=6 yang diperoleh dari range 1-10 dan range 1-100.

Hasil klasifikasi dataset IKG dengan menggunakan algoritma k-NN dengan penentuan nilai *k* secara manual. Hasil rangkuman masing-masing dari 10 kali pengujian diperoleh hasil akurasi pengujian dan disajikan dalam bentuk tabel yang dapat dilihat pada Tabel 5.

Tabel 4. Hasil Pengujian Manual

| Nilai k | Akurasi | Recall | Precission |
|---|---|---|---|
| 3 | 82,82% | 82,32% | 93,10% |
| 5 | 83,26% | 82,04% | 94,48% |
| 7 | 84,14% | 83,03% | 94,48% |
| 15 | 85,02% | 83,23% | 95,86% |
| 17 | 85,46% | 83,33% | 96,55% |
| 19 | 85,90% | 83,83% | 96,55% |
| 23 | 85,46% | 83,33% | 96,55% |
| 24 | 85,90% | 83,83% | 96,55% |
| 27 | 85,46% | 82,94% | 97,24% |
| 30 | 85,02% | 83,23% | 95,86% |

3.1 Uji Beda

Uji beda denga t-Test merupakan sebuah metode pengujian hipotesis menggunakan satu individu (objek penelitian) menggunakan 2 perlakuan yang beda. Meskipun dengan memakai objek sama namun sampel tetap dibagi menajdi dua bagian yaitu data denagn perlakuan 1 (satu) dan data dengan perlakuan 2 (dua). Performa dari masing-masing model dapat diketahui dengan cara membandingkan objek penelitian 1 (satu) dan kondisi objek pada penelitian 2 (dua).

Pengujian t-Test pertama adalah membandingkan seluruh pengujian akurasi dari algoritma k-NN dengan penentuan nilai k menggunakan algoritma genetika dan seluruh hasil akurasi pengujian algoritma k-NN dengan penentuan nilai k secara manual. Uji beda dilakukan dengan menggunakan metode statistik menggunakn tools Microsoft excel 2016.

Untuk menguji hipotesa:
H0 : Tidak terdapat perbedaan nilai akurasi antara model k-NN dengan model GA+k- NN
H1 : Terdapat perbedaan nilai akurasi antara model k-NN dengan model GA+k-NN

Tabel 5 Tabel uji GA+k-NN dengan k-NN

t-Test: Paired Two Sample for Means

| | *k-NN* | *GA+k-NN* |
|---|---|---|
| Mean | 0,84844 | 0,83832 |
| Variance | 0,00011702 | 0,000602526 |
| Observations | 10 | 10 |
| Pearson Correlation | -0,695074384 | |
| Hypothesized Mean Difference | 0 | |
| df | 9 | |
| t Stat | 0,96990885 | |
| P(T<=t) one-tail | 0,178716935 | |
| t Critical one-tail | 1,833112933 | |
| P(T<=t) two-tail | 0,357433869 | |
| t Critical two-tail | 2,262157163 | |

Berdasarkan hasil uji beda dengan t-Test dua sampel berpasangan pada Tabel 6, nilai t hitung yang diwakili oleh nilai t stat sebesar - 0,9699 dan nilai t tabel yang diwakili oleh nilai t critical two tail sebesar 2,26216, maka nilai t hitung < t tabel sehingga H0 diterima dan H1 ditolak. Sedangkan diketahui nilai probabilitas adalah 0,35743 yang mana nilainya tidak < 0,05 yang artinya tidak terdapat perbedaan yang signifikan antara akurasi k-NN dengan GA+k-NN. Namun kombinasi GA+ k-NN tetap mampu memberikan peningkatan akurasi pada dataset indeks kesulitan geografis.

Kemudian tahapan uji beda yang kedua adalah hanya membandingkan nilai akurasi maksimal dari masing-masing metode penentuan nilai *k*, pengujian ini dilakukan dengan *tools* Rapid Miner. Berikut hasil uji beda yang dilakukan dengan menggunakan metode t-Test pada Rapid Miner sehingga diperoleh hasil seperti pada Tabel 7.

Tabel 6. Hasil Uji Beda dengan t-Test

| A | B | C |
|---|---|---|
| | 0,864 +/- 0,052 | 0,860 +/- 0,166 |
| 0,864 +/- 0,052 | | 0,941 |
| 0,860 +/- 0,166 | | |

Probabilities for random values with the same result:

----- 0.941

----- -----

Values smaller than alpha=0.050 indicate a probably significant difference between the mean values!

List of performance values:

0: 0.864 +/- 0.052

1: 0.860 +/- 0.166

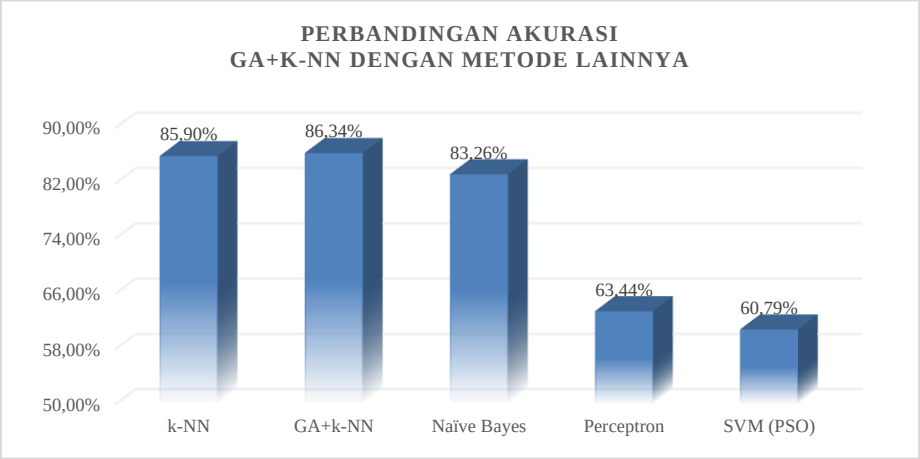
Diperoleh nilai probabilitas adalah 0,941 menghasilkan nilai yang lebih besar dari alpha=0,05 yang mengindikasikan bahwa tidak terdapat perbedaan yang cukup signifikan antara akurasi k-NN dengan GA+kNN namun perpaduan antara algoritma genetika dan k-nearest neighbour tetap menunjukkan perningkatan hasil akurasi pada dataset indeks kesulita geografis sebesar 0,44%.

3.2 Komparasi dengan Metode Lain

Bagian tahapan evaluasi ini juga akan mencoba membandingan hasil klasifikasi Indeks Kesulita dengan beberapa metode atau algoritma yang lain. Metode yang digunakan untuk membandingkan hasil klasifikasi ini adalah dengan *Support Vector Machine (PSO), Perceptron, Naïve Bayes, k-NN, dan GA+k-NN*. Hasil perbandingan dari beberapa metode lainnya ditampilkan pada Tabel 8.

Tabel 7. Perbandingan Nilai Accuracy GA+k-NN dengan Metode Lainnya

| Machine Learning | Accuracy |
|---|---|
| k-NN | 85,90% |
| GA+k-NN | 86,34% |
| Naïve Bayes | 83,26% |
| Perceptron | 63,44% |
| SVM (PSO) | 60,79% |




Gambar 5. Hasil Perbandingan Akurasi GA+k-NN dengan Metode Lain

Grafik perbandingan nilai akurasi GA+k-NN disajikan pada Gambar 6, yang mana nilai akurasi GA+k-NN menunjukkan lebih tinggi dan terbukti lebih unggul daripada metode yang lainnya disusul dengan k-NN manual ditempat kedua dan Naïve Bayes diurutan ketiga.

## 4. KESIMPULAN DAN SARAN

Dari tahapan pengujian antara metode konvensional dan pengujian dengan menggunakan algoritma genetika dalam kasus klasifikasi prioritas bantuan desa dengan menentukan nilai *k* optimal di k-NN menunjukkan hasil yang berbeda. Meskipun tidak ada perbedaan yang cukup signifikan antara k-NN dengan penentuan nilai *k* secara *trial and error* dengan k-NN yang penentuan nilai *k* nya menggunakan algoritma genetika *(GA+k-NN)*, namun optimasi parameter berbasis algoritma genetika untuk menentukan nilai *k* pada algoritma k-NN pada penelitian kali ini mampu memberikan dampak yang cukup baik terhadap hasil model klasifikasi dataset IKG (indeks Kesulitan Geografis), karena dari gabungan algoritma genetika dengan algoritma k-NN (GA+k-NN) ini mampu memberi peningkatan nilai akurasi klasifikasi yang lebih baik sebesar 0,44% jika dibandingkan dengan menggunakan k-NN dengan penentuan nilai *k* secara konvensional. Peningkatan akurasi tersebut diperoleh karena dengan algoritma genetika mampu secara optimal menentukan nilai k, dimana nilai k merupakan bagian terpenting di algoritma k-NN. Selain dapat menentukan nilai k yang optimal, gabungan antara algoritma genetika dan k-NN juga mampu mengurangi kompleksitas dari algoritma k-NN, dikarenakan penghitungan bobot data training sudah tidak dipertimbangkan lagi. Kemudian saat dibandingkan dengan beberapa model lainnya yaitu menggunakan *Naïve Bayes, Perceptron, Support Vector Machine (PSO),*

hasil klasifikasi dataset indeks kesulitas geografis dengan menggunakan GA+k-NN mampu membrikan nilai akurasi yang lebih baik.

## DAFTAR PUSTAKA


[1] J. Han, M. Kamber, and J. Pei, *Introduction*. 2012.

[2] C. Insight, "Dean - Big Data and Data Mining - 2015."

[3] Z. Qin, A. T. Wang, C. Zhang, and S. Zhang, "Cost-Sensitive Classification with k-Nearest Neighbors," pp. 112–131, 2013.

[4] M. Zong, S. Zhang, Y. Zhu, Z. Deng, and D. Cheng, "kNN Algorithm with Data-Driven k Value," pp. 499–512, 2014.

[5] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, 2018.

[6] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, "A novel kNN algorithm with data-driven k parameter computation," *Pattern Recognit. Lett.*, vol. 109, pp. 44–54, 2018.

[7] S. Zhang, "Nearest neighbor selection for iteratively kNN imputation," *J. Syst. Softw.*, vol. 85, no. 11, pp. 2541–2552, 2012.

[8] J. Zierath, R. Rachholz, C. Woernle, and A. Müller, *Load Calculation on Wind Turbines: Validation of Flex5, Alaska/Wind, MSC.Adams and SIMPACK by Means of Field Tests*. 2014.

[9] S. Zhang, X. Wu, and M. Zhu, "Efficient missing data imputation for supervised learning," *Proc. 9th IEEE Int. Conf. Cogn. Informatics, ICCI 2010*, pp. 672–679, 2010.

[10] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," *Water Resour. Res.*, vol. 32, no. 3, pp. 679–693, 1996.

[11] M. STIT, N. Kusuma, and E. Purwanti, "Village Index Analysis Building to Know The Village Development In Gadingrejo District of Pringsewu District," *Inov. Pembang. J. Kelitbangan*, vol. 6, no. 02, pp. 179–190, 2018.

[12] D. A. N. Transmigrasi, "Indeks desa membangun."

[13] H. Harafani, S. Tinggi, M. Informatika, D. Komputer, N. Mandiri, and R. S. Wahono, "Optimasi Parameter pada Support Vector Machine Berbasis Algoritma Genetika untuk Estimasi Kebakaran Hutan," *J. Intell. Syst.*, vol. 1, no. 2, 2015.

[14] N. Harish, S. Mandal, S. Rao, and S. G. Patil, "Particle Swarm Optimization based support vector machine for damage level prediction of non-reshaped berm breakwater," *Appl. Soft Comput. J.*, vol. 27, pp. 313–321, 2015.

[15] L. I. Kuncheva, "Fitness functions in editing k-NN reference set by genetic algorithms," *Pattern Recognit.*, vol. 30, no. 6, pp. 1041–1049, 1997.

[16] N. Suguna and K. Thanushkodi, "An Improved k-Nearest Neighbor Classification Using Genetic Algorithm," *Int. J. Comput. Sci. Issues*, vol. 7, no. 4, pp. 18–21, 2010.

[17] M. North, *Data Mining for the Masses*. 2012.

[18] Z. E. Rasjid and R. Setiawan, "Performance Comparison and Optimization of Text Document Classification using k-NN and Naïve Bayes Classification Techniques," *Procedia Comput. Sci.*, vol. 116, pp. 107–112, 2017.

[19] H. Harafani, T. Informatika, S. Nusa, and M. Jakarta, "OPTIMASI ALGORITMA GENETIKA PADA K-NN UNTUK MEMPREDIKSI KECENDERUNGAN 'BLOG POSTING,'" *J. Pendidik. Teknol. dan Kejuru.*, vol. 15, no. 1, p. 20, 2018.

[20] B. W. Silverman and M. C. Jones, "Estimation Discriminant Analysis Nonparametric Density," vol. 57, no. 3, pp. 233–238, 2014.

[21] P. Bhuvaneswari and A. B. Therese, "Detection of Cancer in Lung with K-NN Classification Using Genetic Algorithm," *Procedia Mater. Sci.*, vol. 10, no. Cnt 2014, pp. 433–440, 2015.

[22] T. Wakahara and Y. Yamashita, "K-NN classification of handwritten characters via

accelerated GAT correlation," *Pattern Recognit.*, vol. 47, no. 3, pp. 994–1001, 2014.
[23] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.
[24] X.-S. Yang, "Chapter 2 - Analysis of Algorithms," pp. 23–44, 2014.
[25] Gorunescu, F. (2011). Intelligent Systems Reference Library. (Gorunescu, Ed)..
[26] L. H. Randy and E. H. Sue, "Practical genetic algorithms," *New York Wiley Sons, Inc*, vol. 50, p. 62, 2004.