

# **LAPORAN PRAKTIKUM *MACHINE LEARNING* MENGGUNAKAN METODE ALGORITMA *K-NEAREST NEIGHBORS* (KNN)**

Diajukan guna untuk memenuhi tugas Kecerdasan Buatan yang dibina  
oleh Lutfi Hakim, S.Pd., M.T.



Oleh :

Ulfiya Nurul Hidayati  
361855401066  
3C

**PROGRAM STUDI DIPLOMA III  
TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BANYUWANGI  
JANUARI 2021**

## I. Judul

Laporan Praktikum *Machine Learning* Menggunakan Metode Algoritma *K-Nearest Neighbors* (KNN)

## II. Tujuan Praktikum

1. Mahasiswa dapat memahami tentang *machine learning*, *supervised learning* dan algoritma KNN (*K-Nearest Neighbors*)
2. Mahasiswa dapat mengaplikasikan metode algoritma KNN (*K-Nearest Neighbors*) untuk menyelesaikan studi kasus pada *machine learning*.

## III. Teori Dasar

### a. *Machine Learning*

*Machine learning* adalah aplikasi dari disiplin ilmu kecerdasan buatan (*Artificial Intelligence*) yang menggunakan teknik statistika untuk menghasilkan suatu model otomatis dari sekumpulan data, dengan tujuan memberikan komputer kemampuan untuk “belajar”. Pembelajaran mesin atau *machine learning* memungkinkan komputer mempelajari sejumlah data (*learn from data*) sehingga dapat menghasilkan suatu model untuk melakukan proses *input-output* tanpa menggunakan kode program yang dibuat secara eksplisit. Proses belajar tersebut menggunakan algoritma khusus yang disebut *machine learning algorithms*. Terdapat banyak algoritma *machine learning* dengan efisiensi dan spesifikasi kasus yang berbeda-beda.

### b. *Supervised Learning*

*Supervised learning* adalah algoritma *machine learning* yang dapat menerapkan informasi yang telah ada pada data dengan memberikan label tertentu, misalnya data klasifikasi sebelumnya (terarah). Algoritma ini mampu memberikan target terhadap *output* yang dilakukan dengan membandingkan pengalaman belajar di masa lalu.

*Supervised Learning* merupakan algoritma yang membangkitkan suatu fungsi yang memetakan *input* ke *output* yang diinginkan. Kualitas hasil pembelajaran sangat bergantung pada kesesuaian *input* dan *output* yang diberikan. Dengan demikian, *user* yang sangat berperan dalam memvalidasi *input* dan *output* tersebut. Oleh karena itu, algoritma jenis ini disebut pembelajaran terawasi (*supervised learning*).

Yang termasuk dalam kategori supervised learning yaitu :

1. Regresi Linier
2. Decision Tree
3. Random Forest
4. k-Nearest Neighbor
5. Support Vector Machine
6. Neural Networks

**c. Algoritma *K-Nearest Neighbor* (KNN)**

*K-Nearest Neighbor* atau yang sering disingkat dengan KNN adalah salah satu algoritma yang digunakan untuk melakukan klasifikasi terhadap objek berdasarkan dari data pembelajaran (data training) yang jaraknya paling dekat dengan objek tersebut.

KNN adalah algoritma *supervised learning* yang maksudnya algoritma ini menggunakan data yang telah ada dan outputnya telah diketahui. KNN banyak dipergunakan pada aplikasi data *mining*, *pattern recognition*, *image processing*, dll.

Algoritma *K-Nearest Neighbor* juga merupakan salah satu algoritma *machine learning* paling sederhana berdasarkan teknik *supervised learning*. Algoritma KNN mengasumsikan kesamaan antara kasus/data baru dan kasus yang tersedia serta memasukkan kasus baru ke dalam kategori yang paling mirip dengan kategori yang tersedia. Konsep algoritma untuk klasifikasi dan regresi dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori K-tetangga terdekat.

**Tujuan dan Penggunaan KKN**

1. Untuk klasifikasi objek baru berdasarkan atribut dan sampel dari data latih/training dengan rumus jarak terdekat.
2. Digunakan dalam aplikasi data mining, statistical pattern recognition, image processing, dll.
3. Beberapa aplikasinya: Pengenalan tulisan tangan, pengenalan emosi/cybersickness berdasarkan aktifitas fisiologis (SpO2, PR, ECG, EEG).

**Kekurangan Algoritma KNN :**

1. Perlu Menunjukkan Parameter K
2. Tidak dapat menangani nilai hilang
3. Sensitif terhadap data pencilan
4. Rentan terhadap variable yang non informatif

### Kelebihan Algoritma KNN :

1. Mudah dipahami dan diimplementasikan
2. Sangat Non Linear

### d. Rumus Jarak dan jenis-jenisnya

#### 1. *Euclidean Distance*/Jarak Euclidean

*Euclidean distance* merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam *Euclidean space* (meliputi bidang *euclidean* dua dimensi, tiga dimensi, atau bahkan lebih). Untuk mengukur tingkat kemiripan data dengan rumus *euclidean distance* maka menggunakan Rumus euclidean Distance dimana akar dari kuadrat perbedaan 2 vektor (root of square differences between 2 vectors).

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

Keterangan :

$d_{ij}$  = tingkat perbedaan (*dissimilarity degree*)

$n$  = jumlah vektor

$x_{ik}$  = vektor citra *input*

$x_{jk}$  = vektor citra pembanding /*output*

**ATAU**

$$d(x, y) = |x - y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

dimana,  $d$  = jarak antara  $x$  dan  $y$

$x$  = data pusat kluster

$y$  = data pada atribut

$i$  = setiap data

$n$  = jumlah data,

$x_i$  = data pada pusat klaster ke  $i$

$y_i$  = data pada setiap data ke  $i$

## 2. *Mahalanobis Distance/ Jarak Mahalanobis*

Penghitungan jarak Mahalanobis ini didasarkan pada korelasi antara variabel dan dengan pola yang berbeda dapat diidentifikasi dan dianalisis berdasarkan titik referensinya.

$$d(x, y) = -\sum_i^n z_i x_i y_i$$

Di mana nilai  $Z_i$

$$Z_i = \frac{\lambda_i}{\sqrt{\lambda_i + \alpha^2}}$$

$\lambda_i$  bergantung pada

$$\alpha = 0.25$$

$\lambda_i$  bergantung pada besarnya eigenvalues

## 3. *Manhattan Distance/Jarak Manhattan*

*Manhattan distance* digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat sepasang objek. Rumus yang digunakan adalah sebagai berikut:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

dimana,  $d$  = jarak antara  $x$  dan  $y$

$x$  = data pusat kluster

$y$  = data pada atribut

$i$  = setiap data

$n$  = jumlah data,

$x_i$  = data pada pusat klaster ke  $i$

$y_i$  = data pada setiap data ke  $i$

## ATAU

*City Block Distance* yang juga dikenal sebagai *Manhattan Distance*, *Boxcar Distance*, *Absolute Value Distance*, merepresentasikan jarak antara dua objek secara absolut, dengan formula sebagai berikut:

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}|$$

Contoh:

Jika A (0,3,4,5) dan B (7,6,3,-1), maka *city block distance* antara A dan B adalah :  $d_{BA} = |0-7| + |3-6| + |4-3| + |5+1|$

$$= 7+3+1+6$$

$$= 17$$

#### 4. *Canberra Distance/ Jarak Canberra*

Untuk setiap nilai 2 vektor yang akan dicocokkan, Canberra Distance membagi absolute selisih 2 nilai dengan jumlah dari absolute 2 nilai tersebut[3]. Hasil dari dua nilai yang dicocokkan lalu dijumlahkan untuk mendapatkan Canberra Distance. Jika koordinat nol-nol((0,0)) diberikan definisi dengan 0/0=0. Canberra Distance ini sangat peka terhadap sedikit perubahan dengan kedua koordinat mendekati nol.

$$d_{ij} = \sum_{k=1}^n \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}$$

Keterangan :

$D_{ij}$  = tingkat perbedaan (*dissimilarity degree*)

$n$  = jumlah vektor

$X_{ik}$  = vektor citra *input*

$X_{jk}$  = vektor citra pembanding */output*

## ATAU

menghitung penjumlahan dari pecahan-pecahan perbedaan kordinat antara dua buah titik. Setiap pecahan bernilai antara 0 hingga 1. Formulanya adalah sebagai berikut :

$$d_{ij} = \sum_{k=1}^n \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}$$

Contoh:

Jika A (0,3,4,5) dan B (7,6,3,-1), maka *canberra distance* antara A dan B adalah :

$$\begin{aligned} d_{BA} &= \frac{|0-7|}{0+7} + \frac{|3-6|}{3+6} + \frac{|4-3|}{4+3} + \frac{|5+1|}{5+1} \\ &= 1 + 1/3 + 1/7 + 1 \\ &= 2,476 \end{aligned}$$

## 5. *Minkowski Distance*/ Jarak Minkowski

*Minkowski distance* merupakan sebuah metrik dalam ruang vektor di mana suatu norma didefinisikan (*normed vector space*) sekaligus dianggap sebagai generalisasi dari *Euclidean distance* dan *Manhattan distance*. Dalam pengukuran jarak objek menggunakan *minkowski distance* biasanya digunakan nilai  $p$  adalah 1 atau 2. Berikut rumus yang digunakan menghitung jarak dalam metode ini.

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

dimana,  $d$  = jarak antara  $x$  dan  $y$

$x$  = data pusat klaster

$y$  = data pada atribut

$i$  = setiap data

$n$  = jumlah data,

$x_i$  = data pada pusat klaster ke  $i$

$y_i$  = data pada setiap data ke  $i$

$p$  = power

### ATAU

Minkowski Distance merupakan penjabaran dari *Euclidean*, *City Block*, dan *Chebyshev*. *Minkowski* menggunakan bilangan *order*  $\lambda$ . Apabila  $\lambda=1$ , maka merupakan *City*

*Block Distance*. Apabila  $\lambda=2$ , maka merupakan *Euclidean Distance*. Apabila  $\lambda=\infty$ , maka merupakan *Chebyshev Distance*. Formula dari *Minkowski Distance* adalah sebagai berikut :

$$d_{ij}^{\lambda} = \sqrt[p]{\sum_{k=1}^n (x_{ik} - x_{jk})^{\lambda}}$$

Contoh:

Jika A (0,3,4,5) dan B (7,6,3,-1), maka *minkowski distance* dengan order 3 antara A dan B adalah :  $d_{BA} = ((0-7)^3 + (3-6)^3 + (4-3)^3 + (5+1)^3)^{1/3}$

$$= (-343-27+1+216)^{1/3}$$

$$= (-533)^{1/3}$$

$$= -5,348$$

(Jarak antara B dan A ( $d_{AB}$ ) adalah 5,348, berbeda dengan jarak  $d_{BA}$ . Biasanya yang digunakan adalah yang bernilai positif).

## 6. *Hamming Distance/ Jarak Hamming*

Hamming distance adalah cara untuk mengukur jarak antara dua string yang ukurannya sama dengan membandingkan simbol-simbol yang terdapat pada kedua string pada posisi yang sama. Metode hamming distance merupakan salah satu dari algoritma approximate string matching yang ditemukan oleh Richard Hamming, pada tahun 1950. Metode hamming distance pertama kali digunakan untuk mendeteksi dan memperbaiki telekomunikasi sebagai estimasi error. Hamming distance dari dua string adalah jumlah simbol dari kedua string yang berbeda. Sebagai contoh Hamming Distance antara string "karolin" dan "kerstin" adalah 3. Hamming distance juga dapat digunakan untuk mengukur jarak antar dua string binary misalnya jarak antara "1011101" dengan "1001001" adalah 2.



$$d_{ij} = q + r$$

Dimana  $q$  adalah jumlah variabel dengan nilai 1 pada objek ke- $i$  tapi bernilai 0 pada objek ke- $j$ . Sedangkan  $r$  adalah jumlah variabel dengan nilai 0 pada objek ke- $i$  tapi bernilai 1 pada objek ke- $j$ .

## 7. *Chebyshev Distance*/Jarak Chebyshev

*Chebyshev Distance* disebut juga *Maximum Value Distance* menghitung *max* dari besaran absolut perbedaan antara koordinat dari sepasang objek. Formulanya adalah sebagai berikut :

$$d_{ij}^c = \max_k |x_{ik} - x_{jk}|$$

Contoh:

Jika A (0,3,4,5) dan B (7,6,3,-1), maka *chebyshev distance* antara A dan B adalah :  
 $d_{BA} = \max \{ |0-7|, |3-6|, |4-3|, |5+1| \} = \max \{ 7, 3, 1, 6 \} = 7$

### e. Runtutan Algoritma KNN

1. Menentukan parameter  $k$  (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak *eucliden* objek terhadap data *training* yang diberikan.
3. Mengurutkan hasil no 2 secara *ascending* (berurutan dari nilai tinggi ke rendah)
4. Mengumpulkan kategori  $Y$  (Klasifikasi *nearest neighbor* berdasarkan nilai  $k$ )
5. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka dapat diprediksikan kategori objek.

## IV. Source Code

### a. Source Code DataLatih.java

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package PraktikumKNN;
7
8  /**
9   *
10   * @author Alkaafi Komputer
11   */
12  public class DataLatih {
13
14      // Data Latih
15      int[] NR = {5, 4, 9, 6, 10, 9, 3, 5, 9, 2};
16      int[] UN = {6, 6, 8, 5, 8, 6, 4, 3, 4, 3};
17      String[] kelas = {"REGULER", "REGULER", "BEASISWA", "REGULER",
18                       "BEASISWA", "BEASISWA", "TIDAK LULUS", "TIDAK LULUS", "REGULER", "TIDAK LULUS"};
19  }
```

### b. Source Code KNN\_algoritma.java

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package PraktikumKNN;
7
8  import java.io.IOException;
9  import java.util.HashMap;
10 import java.util.Iterator;
11 import java.util.Map;
12 import java.util.Set;
13 import java.util.TreeMap;
14
15 /**
16  *
17  * @author Alkaafi Komputer
18  */
19 public class KNN_algoritma {
20     public static String ranking = "";
21     public String knn(int nR, int uN) throws IOException{
22
23         // Data Training
24         DataLatih latih = new DataLatih();
25
26         // Data Testing
27         int NR = nR;
28         int UN = uN;
29
30         double hasiled, hasiled_before;
```

```

31      Map knn = new HashMap<>();
32      System.out.println("DATA TESTING");
33      System.out.println("NR : " + NR);
34      System.out.println("UN : " + UN);
35      System.out.println("");
36      System.out.println("Hasil Jarak Euclidean");
37      System.out.println("Nilai Rapot \t" + "Nilai UN \t" + "Kelas \t" + "\t ED");
38
39      for(int a = 1; a<10; a++){
40          hasilED = jarakEuclidean(latih.NR[a], NR, latih.UN[a], UN);
41          System.out.println(latih.NR[a] + "\t \t" + latih.UN[a] + "\t \t" + latih.kelas[a] + "\t \t" + hasilED);
42          knn.put(hasilED, latih.kelas[a]);
43      }
44
45      System.out.println("Ranking Euclidean Distance");
46      Map knn_hasil = new TreeMap(knn);
47      printMap(knn_hasil);
48
49
50      System.out.println("");
51      System.out.println("Hasil Keputusan : " + knn_hasil.values().toArray()[0]);
52      return (String) knn_hasil.values().toArray()[0];
53  }
54
55  public static void printMap(Map map) {
56      Set s = map.entrySet();
57      Iterator it = s.iterator();
58      int i = 0;
59      while (it.hasNext()) {
60          Map.Entry entry = (Map.Entry) it.next();
61          Double key = (double) entry.getKey();
62          String value = (String) entry.getValue();
63          ranking = ranking + "\n " + "Apel: " + i + " nilai: " + key + " => " + value;
64          System.out.println(key + " => " + value);
65          i++;
66      }
67      System.out.println("=====");
68  }
69
70  // Membuat function untuk jarak Euclidean
71  static double jarakEuclidean(int R1, int R2, int G1, int G2) {
72      return Math.sqrt(Math.pow(R1 - R2, 2) + Math.pow(G1 - G2, 2));
73  }
74  }

```

### c. Source Code KNN\_run.java

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package PraktikumKNN;
7
8  import java.io.IOException;
9
10 /**
11  *
12  * @author Alkaafi Komputer
13  */
14 public class KNN_run {
15     /**
16     * @param args the command line arguments
17     */
18
19     /**
20     *
21     */
22     public static void main(String[] args) throws IOException {
23         // TODO code application logic here
24         //Data Testing
25
26         int nr = 6;
27         int un = 9;
28
29         KNN_algoritma deteksi = new KNN_algoritma();
30         String hasil = null;
31
32         hasil = deteksi.knn(nr, un);
33     }
34 }

```

## V. Hasil (Output Program dari KNN)

```
Output - 3C_361855401066_Ulfiya Nurul Hidayati_PrakAI (run)

run:
DATA TESTING
NR : 6
UN : 9

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4              6          REGULER  3.605551275463989
9              8          BEASISWA 3.1622776601683795
6              5          REGULER  4.0
10             8          BEASISWA 4.123105625617661
9              6          BEASISWA 4.242640687119285
3              4          TIDAK LULUS 5.830951894845301
5              3          TIDAK LULUS 6.082762530298219
9              4          REGULER  5.830951894845301
2              3          TIDAK LULUS 7.211102550927978

Ranking Euclidean Distance
3.1622776601683795 => BEASISWA
3.605551275463989 => REGULER
4.0 => REGULER
4.123105625617661 => BEASISWA
4.242640687119285 => BEASISWA
5.830951894845301 => REGULER
6.082762530298219 => TIDAK LULUS
7.211102550927978 => TIDAK LULUS
=====

Hasil Keputusan : BEASISWA
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Analisa :** pada output percobaan diatas ditunjukkan bahwa data testing dengan Nilai Raport 6 dan nilai UN 9, maka hasil keputusan menunjukkan bahwa anak tersebut masuk kedalam tingkat mendapatkan Beasiswa. Hal ini dikarenakan kedua acuan nilai didalam menggunakan rumus jarak Euclidean dan menggunakan metode KNN.

## VI. Eksperimen dan Pembahasan

### a. Hasil Percobaan 1 (NR : 8 dan UN : 10)

```
Output - 3C_361855401066_Ulfiya Nurul Hidayati_PrakAI (run)

DATA TESTING
NR : 8
UN : 10

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4              6          REGULER  5.656854249492381
9              8          BEASISWA 2.23606797749979
6              5          REGULER  5.385164807134504
10             8          BEASISWA 2.8284271247461903
9              6          BEASISWA 4.123105625617661
3              4          TIDAK LULUS 7.810249675906654
5              3          TIDAK LULUS 7.615773105863909
9              4          REGULER  6.082762530298219
2              3          TIDAK LULUS 9.219544457292887

Ranking Euclidean Distance
2.23606797749979 => BEASISWA
2.8284271247461903 => BEASISWA
4.123105625617661 => BEASISWA
5.385164807134504 => REGULER
5.656854249492381 => REGULER
6.082762530298219 => REGULER
7.615773105863909 => TIDAK LULUS
7.810249675906654 => TIDAK LULUS
9.219544457292887 => TIDAK LULUS
=====

Hasil Keputusan : BEASISWA
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Analisa :** pada output percobaan 1 ditunjukkan bahwa data testing dengan Nilai Raport 8 dan nilai UN 10, maka hasil keputusan menunjukkan bahwa anak tersebut masuk kedalam tingkat mendapatkan Beasiswa. Hal ini dikarenakan kedua acuan nilai didalam menggunakan rumus jarak Euclidean dan menggunakan metode KNN.

#### b. Hasil Percobaan 2 (NR : 10 dan UN : 5)

```
Output - 3C_361855401066_Ulfiya Nurul Hidayati_PrakAI (run)

Run:
DATA TESTING
NR : 10
UN : 5

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4              6          REGULER  6.082762530298219
9              8          BEASISWA 3.1622776601683795
6              5          REGULER  4.0
10             8          BEASISWA 3.0
9              6          BEASISWA 1.4142135623730951
3              4          TIDAK LULUS 7.0710678118654755
5              3          TIDAK LULUS 5.385164807134504
9              4          REGULER  1.4142135623730951
2              3          TIDAK LULUS 8.246211251235321

Ranking Euclidean Distance
1.4142135623730951 => REGULER
3.0 => BEASISWA
3.1622776601683795 => BEASISWA
4.0 => REGULER
5.385164807134504 => TIDAK LULUS
6.082762530298219 => REGULER
7.0710678118654755 => TIDAK LULUS
8.246211251235321 => TIDAK LULUS

=====

Hasil Keputusan : REGULER
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Analisa :** pada output percobaan 2 ditunjukkan bahwa data testing dengan Nilai Raport 10 dan nilai UN 5, maka hasil keputusan menunjukkan bahwa anak tersebut masuk kedalam tingkat Reguler. Hal ini dikarenakan menggunakan rumus jarak Euclidean dan menggunakan metode KNN.

#### c. Hasil Percobaan 3 (NR : 3 dan UN : 5)

```
Output - 3C_361855401066_Ulfiya Nurul Hidayati_PrakAI (run)

Run:
DATA TESTING
NR : 3
UN : 5

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4              6          REGULER  1.4142135623730951
9              8          BEASISWA 6.708203932499369
6              5          REGULER  3.0
10             8          BEASISWA 7.615773105863909
9              6          BEASISWA 6.082762530298219
3              4          TIDAK LULUS 1.0
5              3          TIDAK LULUS 2.8284271247461903
9              4          REGULER  6.082762530298219
2              3          TIDAK LULUS 2.23606797749979

Ranking Euclidean Distance
1.0 => TIDAK LULUS
1.4142135623730951 => REGULER
2.23606797749979 => TIDAK LULUS
2.8284271247461903 => TIDAK LULUS
3.0 => REGULER
6.082762530298219 => REGULER
6.708203932499369 => BEASISWA
7.615773105863909 => BEASISWA

=====

Hasil Keputusan : TIDAK LULUS
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Analisa :** pada output percobaan 3 ditunjukkan bahwa data testing dengan Nilai Raport 3 dan nilai UN 5, maka hasil keputusan menunjukkan bahwa nilai-nilai tersebut masuk kedalam tingkat Tidak Lulus atau dinyatakan Tidak Lulus. Hal ini dikarenakan menggunakan rumus jarak Euclidean dan menggunakan metode KNN.

## VII. Kesimpulan

Pada praktikum kali ini mencoba mengaplikasikan menggunakan metode KNN (*K-Nearest Neighbor*). Algoritma *K-Nearest Neighbor* merupakan salah satu algoritma *machine learning* paling sederhana berdasarkan teknik *supervised learning*. Algoritma KNN mengasumsikan kesamaan antara kasus atau data baru dan kasus yang tersedia serta memasukkan kasus baru ke dalam kategori yang paling mirip dengan kategori yang tersedia. Konsep algoritma untuk klasifikasi dan regresi dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori *K*-tetangga terdekat. Algoritma *K-Nearest Neighbor* memiliki banyak rumus jarak salah satunya *Euclidean distance*. *Euclidean distance* merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam *Euclidean space* (meliputi bidang *euclidean* dua dimensi, tiga dimensi, atau bahkan lebih). Untuk mengukur tingkat kemiripan data dengan rumus *euclidean distance* maka menggunakan rumus *euclidean Distance* dimana akar dari kuadrat perbedaan 2 vektor.

## VIII. Referensi

- Advernesia. Apa itu Machine Learning dan Cara Kerjanya <https://www.advernesia.com/blog/data-science/machine-learning-adalah/>. Diakses 20 Januari 2021
- Maulana, A. Supervised, Unsupervised, Semi-Supervised learning  
<http://arofiqimaulana.com/supervised-unsupervised-semi-supervised-learning/>. Diakses 20 Januari 2021
- Ketutrare. Algoritma K-Nearest Neighbor dan Contoh Soal  
<https://www.ketutrare.com/2018/11/algoritma-k-nearest-neighbor-dan-contoh-soal.html>.  
Diakses 20 Januari 2021
- Informatikalogi. Algoritma K-Nearest Neighbor (K-NN) <https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor/>. Diakses 21 Januari 2021
- HAKIM, L. (2021). Modul Machine Learning. KECERDASAN BUATAN-TEKNIK INFORMATIKA POLITEKNIK NEGERI BANYUWANGI.  
<https://elearning.poliwangi.ac.id/mod/resource/view.php?id=4589>. Diakses 19 Januari 2021

Wurdianarto, Shendy Rachmat. Dkk. PERBANDINGAN EUCLIDEAN DISTANCE DENGAN CANBERRA DISTANCE PADA FACE RECOGNITION. Publikasi.dinus.ac.id. Diakses 21 januari 2021.

Nishom, M. Perbandingan Akurasi *Euclidean Distance*, *Minkowski Distance*, dan *Manhattan Distance* pada Algoritma *KMeans Clustering* berbasis *Chi-Square*. Ejournal.poltektegal.ac.id. Diakses 21 januari 2021.

Swedia, Ericks Rahmat. Dkk. Aplikasi Pendeteksi Rambu Lalu-Lintas Menggunakan Operator Sobel dan Metode Hamming. Journal.uui.ac.id. Diakses 21 januari 2021.

Murti, Darlis Heru.dkk. *CLUSTERING DATA NON-NUMERIK DENGAN PENDEKATAN ALGORITMA K-MEANS DAN HAMMING DISTANCE* STUDI KASUS BIRO JODOH. Juti.if.its.ac.id. Diakses 21 januari 2021.