

LAPORAN PRAKTIKUM MACHINE LEARNING DENGAN MENGGUNAKAN METODE ALGORITMA K-NEAREST NEIGHBORS

Laporan

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



Oleh :

Arum Indah Sari
361855401078
3C

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

I. Judul

Laporan Praktikum Machine Learning Dengan Menggunakan Metode Algoritma K-Nearest Neighbors

II. Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami tentang Machine Learning dan Supervised Learning
2. Mahasiswa dapat mengetahui dan memahami tentang Algoritma KNN
3. Mahasiswa dapat menyelesaikan studi kasus Machine Learning menggunakan metode Algoritma KNN

III. Teori Dasar

➤ Machine Learning

Machine Learning merupakan salah satu cabang dari ilmu Kecerdasan Buatan, khususnya yang mempelajari tentang bagaimana komputer mampu belajar dari data untuk meningkatkan kecerdasannya. Pembelajaran mesin dikembangkan berdasarkan disiplin ilmu lainnya seperti statistika, matematika dan *data mining* sehingga mesin dapat belajar dengan menganalisa data tanpa perlu di program ulang atau diperintah.

Istilah *machine learning* pertama kali dikemukakan oleh beberapa ilmuwan matematika seperti Adrien Marie Legendre, Thomas Bayes dan Andrey Markov pada tahun 1920-an dengan mengemukakan dasar-dasar *machine learning* dan konsepnya. Sejak saat itu ML banyak yang mengembangkan.

Berikut ini adalah pengertian Machine Learning menurut beberapa ahli.

➤ Menurut Arthur (1959):

Kemampuan komputer untuk melakukan pembelajaran tanpa harus menjelaskan atau terprogram secara eksplisit kepada komputer.

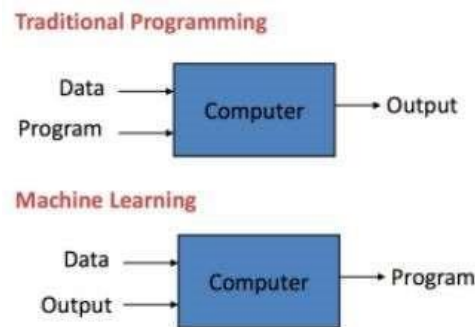
➤ Menurut Tom Mitchel (1997)

Komputer yang memiliki kemampuan melakukan belajar dari pengalaman terhadap tugas-tugasnya dan mengalami peningkatan kinerja.

➤ Menurut Budiharto (2016)

Tipe dari kecerdasan buatan yang menyediakan komputer dengan kemampuan untuk belajar dari data, tanpa secara eksplisit harus mengikuti instruksi terprogram.

Dengan melihat berbagai definisi di atas, *Machine learning* memiliki fokus pada pengembangan sebuah sistem yang mampu belajar sendiri untuk memutuskan sesuatu, tanpa harus berulang kali diprogram oleh manusia. Dengan metode tersebut, mesin tidak hanya bisa menemukan aturan untuk perilaku optimal dalam pengambilan keputusan, namun juga bisa beradaptasi dengan perubahan yang terjadi. Dalam pembelajaran mesin, dapat menganalisis kumpulan data yang besar untuk menemukan pola.



Gambar Perbedaan Pemrograman Tradisional dan Machine Learning
(Brownlee, 2015)

Melalui algoritma pembelajaran mesin, pengguna perlu melatih komputer sedemikian rupa sehingga bisa memahami model objek yang dikenali manusia. Lalu apa perbedaan pemrograman tradisional dengan pemrograman menggunakan machine learning? Gambar diatas menunjukkan perbedaan keduanya menurut Brownlee (2015). Pada pemrograman tradisional, data dan program dijalankan pada komputer untuk menghasilkan output, sedangkan pada pemrograman menggunakan *machine learning*, data dan output dijalankan pada komputer untuk membuat program, dan kemudian program tersebut bisa digunakan dalam pemrograman tradisional.

Dalam hal ini *machine learning* memiliki kemampuan untuk memperoleh data yang ada dengan perintah ia sendiri. ML juga dapat mempelajari data yang ada dan data yang ia peroleh sehingga bisa melakukan tugas tertentu. Tugas yang dapat dilakukan oleh ML pun sangat beragam, tergantung dari apa yang ia pelajari.

Salah satu contoh dari penerapan ML yang cukup terkenal adalah Deep Blue yang dibuat oleh IBM pada tahun 1996. Deep Blue merupakan *machine learning* yang dikembangkan

agar bisa belajar dan bermain catur. Deep Blue juga telah diuji coba dengan bermain catur melawan juara catur profesional dan Deep Blue berhasil memenangkan pertandingan catur tersebut.

Peran *machine learning* banyak membantu manusia dalam berbagai bidang. Bahkan saat ini penerapan ML dapat dengan mudah kamu temukan dalam kehidupan sehari-hari. Misalnya saat menggunakan fitur *face unlock* untuk membuka perangkat *smartphone*, atau saat menjelajah di internet atau media sosial pengguna akan sering disuguhkan dengan beberapa iklan. Iklan-iklan yang dimunculkan juga merupakan hasil pengolahan ML yang akan memberikan iklan sesuai dengan pribadi pengguna.

Sebenarnya masih banyak contoh dari penerapan *machine learning* yang sering dijumpai. Lalu pertanyaannya, bagaimana ML dapat belajar? ML bisa belajar dan menganalisa data berdasarkan data yang diberikan saat awal pengembangan dan data saat ML sudah digunakan. ML akan bekerja sesuai dengan teknik atau metode yang digunakan saat pengembangan. Ada beberapa teknik yang dimiliki oleh *machine learning*, namun secara luas ML memiliki dua teknik dasar belajar, yaitu *supervised* dan *unsupervised*.

➤ **Supervised Learning**

Metode *Supervised Learning* merupakan metode yang bisa diterapkan pada pembelajaran mesin yang bisa menerima informasi yang sudah ada pada data dengan memberikan label tertentu. Diharapkan metode ini bisa memberikan target terhadap *output* yang dilakukan dengan membandingkan pengalaman belajar di masa lalu. Karenanya, kelemahan dari Supervised Learning yang paling pokok adalah pengguna harus merumuskan output yang benar. Jika proses pelatihan dilakukan dengan benar, maka tentu nantinya mesin akan melakukan eksekusi dengan benar. Terdapat Dua jenis supervised learning yaitu:
Regresi : melakukan prediksi dengan nilai output berupa bilangan kontinu
Klasifikasi : memprediksi nilai output dalam bentuk data kategori.

Contoh Regresi :

- Memprediksi harga jual rumah berdasarkan luas tanah dan usia bangunan.
- Menentukan tinggi pohon berdasarkan diameter batang
- Memprediksi jumlah penjualan barang berdasarkan durasi iklan di televisi, jumlah promosi di media social, jumlah pamflet yang disebar.

Beberapa jenis regresi:

1. Regresi Linear: menggunakan 1 atau lebih variable bergantung dengan asumsi fungsi yang digunakan untuk memprediksi adalah fungsi linear.
2. Regresi Nonlinear: menggunakan 1 atau lebih variable bergantung dengan asumsi fungsi yang digunakan untuk memprediksi adalah fungsi non linear.

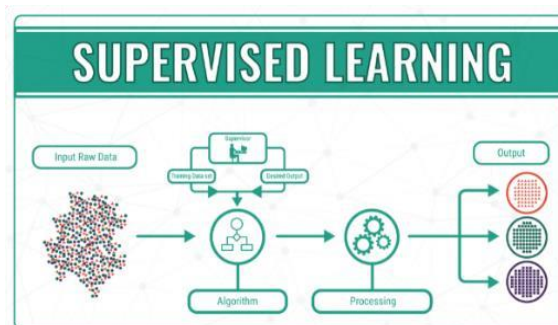
Contoh Klasifikasi :

- Pengelompokkan email spam dan bukan spam
- Klasifikasi sentiment positif, negative dan netral
- Pengenalan objek kendaraan mobil, sepeda, dan sepeda motor

Klasifikasi terbagi 2:

1. Binary Classes : Klasifikasi dengan 2 jenis kelas seperti spam atau bukan spam, positif atau negative, mahal atau murah.
2. Multi Classes : Klasifikasi dengan lebih dari 2 kelas seperti klasifikasi jenis kendaraan motor, pengenalan ucapan, pengenalan wajah.

Langkah Kerja algoritma ini yaitu sistem diberikan *training data set* berupa informasi input dan output yang diinginkan sehingga sistem akan mempelajari berdasarkan data yang telah ada, sistem akan mencari pola dari *data set* sehingga pola tersebut akan dijadiakannya sebagai acuan untuk data-data berikutnya, dibawah ini adalah cara kerja dari *Supervised Learning*.



Gambar Langkah Kerja *Supervised Learning*

➤ Algoritma KNN

Algoritma k-NN adalah suatu metode yang menggunakan algoritma supervised. Perbedaan antara supervised learning dengan unsupervised learning adalah pada supervised learning bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada unsupervised learning, data belum memiliki pola apapun, dan tujuan unsupervised learning untuk menemukan pola dalam sebuah data. Tujuan dari algoritma k-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training samples.

K-Nearest Neighbor sering digunakan dalam klasifikasi dengan tujuan dari algoritma ini adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training samples. Algoritma K-Nearest Neighbor (K-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Teknik ini sangat sederhana dan mudah diimplementasikan. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean.

Untuk mendefinisikan jarak antara dua titik yaitu titik pada *data training* (x) dan titik pada *data testing* (y) maka digunakan rumus *Euclidean* sebagai berikut :

$$\sum_{i=1}^n f(T_i, S_i) * W_i$$
$$similarity(T, S) = W_i$$

Keterangan :

T : kasus baru

S : kasus yang ada dalam penyimpanan n : jumlah atribut dalam setiap kasus i : atribut individu antara 1 s.d. n f : fungsi *similarity* atribut I antara kasus T dengan kasus S w : bobot yang diberikan pada atribut ke- i .

➤ 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 kluster 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.

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

Di mana nilai Z_i $Z_i = \sqrt{\frac{\lambda_i}{\lambda_i + \alpha^2}}$
 λ_i bergantung pada:

$$\alpha = 0.25$$

λ_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 kluster 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* λ . 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 :

$$d_y^j = \sqrt[n]{\sum_{k=1}^n (x_{ik} - x_{jk})^n}$$

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 kordinat dari sepasang objek. Formulasnya adalah sebagai berikut :

$$d_y^j = \max_k |x_{ik} - x_{jk}|$$

Contoh:

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

➤ Runtutan Algoritma KNN

1. Menentukan parameter k (jumlah banyaknya tetangga terdekat)
2. Menghitung jarak antara data baru dan semua data yang ada di data training
3. Mengurutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke – k
4. Menentukan kategori dari tetangga terdekat
5. Menggunakan kategori mayoritas yang sederhana dari tetangga mana yang terdekat tersebut sebagai nilai prediksi yang baru

IV. Source Code Algoritma KNN

➤ 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 arum indah sari
11  */
12  public class DataLatih {
13      // Data Latih
14      int[] NR = {5, 4, 9, 6, 10, 9, 3, 5, 9, 2};
15      int[] UN = {6, 6, 8, 5, 8, 6, 4, 3, 4, 3};
16      String[] kelas = {"REGULER", "REGULER", "BEASISWA", "REGULER",
17                       "BEASISWA", "BEASISWA", "TIDAK LULUS", "TIDAK LULUS", "REGULER", "TIDAK LULUS"};
18  }
19
```

➤ KNN_run.java

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

➤ 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 arum indah sari
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
32         Map knn = new HashMap();
33         System.out.println("DATA TESTING");
34         System.out.println("NR : " + NR);
35         System.out.println("UN : " + UN);
36         System.out.println("");
37         System.out.println("Hasil Jarak Euclidean");
38         System.out.println("Nilai Rapot \t" + "Nilai UN \t" + "Kelas \t" + "\t ED");
39
40         for(int a = 1; a<10; a++){
41             hasilED = jarakEuclidean(latih.NR[a], NR, latih.UN[a], UN);
42             System.out.println(latih.NR[a] + "\t\t" + latih.UN[a] + "\t\t" + latih.kelas[a] + "\t\t" + hasilED);
43             knn.put(hasilED, latih.kelas[a]);
44         }
45
46         System.out.println("Ranking Euclidean Distance");
47         Map knn_hasil = new TreeMap(knn);
48         printMap(knn_hasil);
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;
62             key = (Double) entry.getKey();
63             String value = (String) entry.getValue();
64             ranking = ranking + "\n " + "Apel: " + i + " nilai: " + key + " => " + value;
65             System.out.println(key + " => " + value);
66             i++;
67         }
68         System.out.println("=====");
69     }
70
71     // Membuat function untuk jarak Euclidean
72     static double jarakEuclidean(int R1, int R2, int G1, int G2) {
73         return Math.sqrt(Math.pow(R1 - R2, 2) + Math.pow(G1 - G2, 2));
74     }
75 }

```

V. Hasil/Output Algoritma KNN

➤ Hasil Percobaan 1

```
Output - PraktikumKNN (run) x Start Page x DataLatih.java x KNN_run.java x
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)
```

Pada output percobaan diatas ditunjukkan bahwa murid atau mahasiswa tersebut masuk kelas Beasiswa, hal ini dikarenakan Data Testing yaitu Nilai Rapot bernilai 6 dan Ujian Nasional bernilai 9. Maka dengan menggunakan rumus jarak Euclidean dan dengan menggunakan metode algoritma KNN, maka hasil keputusan menunjukkan kelas Beasiswa.

➤ Hasil Percobaan 2

```
Output - PraktikumKNN (run) x Start Page x DataLatih.java x KNN_run.java x KNN_algoritma.java x
run:
DATA TESTING
NR : 5
UN : 4

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4              6          REGULER  2.23606797749979
9              8          BEASISWA 5.656854249492381
6              5          REGULER  1.4142135623730951
10             8          BEASISWA 6.4031242374328485
9              6          BEASISWA 4.47213595499958
3              4          TIDAK LULUS 2.0
5              3          TIDAK LULUS 1.0
9              4          REGULER  4.0
2              3          TIDAK LULUS 3.1622776601683795

Ranking Euclidean Distance
1.0 => TIDAK LULUS
1.4142135623730951 => REGULER
2.0 => TIDAK LULUS
2.23606797749979 => REGULER
3.1622776601683795 => TIDAK LULUS
4.0 => REGULER
4.47213595499958 => BEASISWA
5.656854249492381 => BEASISWA
6.4031242374328485 => BEASISWA

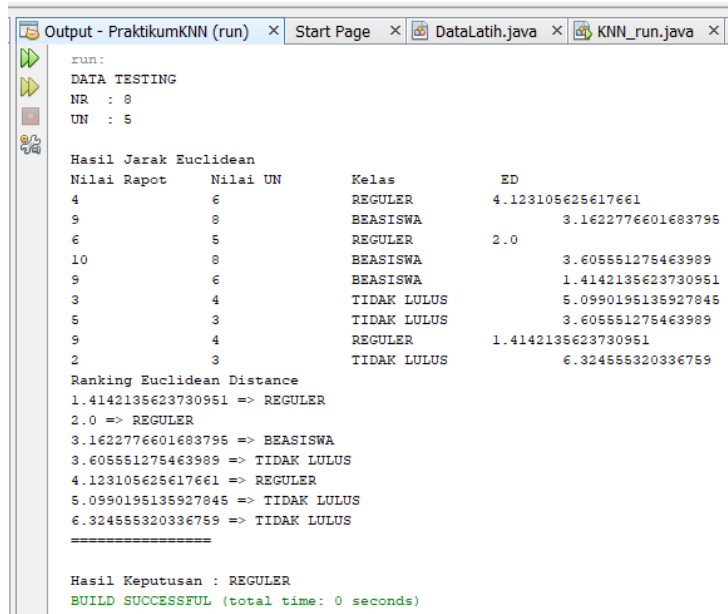
=====

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

Pada output percobaan diatas ditunjukkan bahwa murid atau mahasiswa tersebut masuk kelas Tidak Lulus, hal ini dikarenakan Data Testing yaitu Nilai Rapot bernilai 5 dan Ujian Nasional bernilai 4. Maka dengan menggunakan rumus jarak Euclidean dan

dengan menggunakan metode algoritma KNN, maka hasil keputusan menunjukkan kelas Tidak Lulus.

➤ Hasil Percobaan 3



```
run:
DATA TESTING
NR : 8
UN : 5

Hasil Jarak Euclidean
Nilai Rapot    Nilai UN    Kelas    ED
4             6         REGULER  4.123105625617661
9             8         BEASISWA 3.1622776601683795
6             5         REGULER  2.0
10            8         BEASISWA 3.605551275463989
9             6         BEASISWA 1.4142135623730951
3             4         TIDAK LULUS 5.0990195135927845
5             3         TIDAK LULUS 3.605551275463989
9             4         REGULER  1.4142135623730951
2             3         TIDAK LULUS 6.324555320336759

Ranking Euclidean Distance
1.4142135623730951 => REGULER
2.0 => REGULER
3.1622776601683795 => BEASISWA
3.605551275463989 => TIDAK LULUS
4.123105625617661 => REGULER
5.0990195135927845 => TIDAK LULUS
6.324555320336759 => TIDAK LULUS
=====

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

Pada output percobaan diatas ditunjukkan bahwa murid atau mahasiswa tersebut masuk kelas Beasiswa, hal ini dikarenakan Data Testing yaitu Nilai Rapot bernilai 8 dan Ujian Nasional bernilai 5. Maka dengan menggunakan rumus jarak Euclidean dan dengan menggunakan metode algoritma KNN, maka hasil keputusan menunjukkan kelas Reguler.

VI. Kesimpulan

Pada praktikum Kecerdasan Buatan kali ini mencoba untuk mengetahui nilai dengan melakukan percobaan dengan Metode Algoritma KNN atau Algoritma K-Nearest Neighbors. Algoritma k-NN adalah suatu metode yang menggunakan algoritma supervised. Tujuan dari algoritma k-NN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training samples.

K-Nearest Neighbor sering digunakan dalam klasifikasi dengan tujuan dari algoritma ini adalah untuk mengklasifikasi objek baru berdasarkan atribut dan training samples. Algoritma K-Nearest Neighbor (K-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Algoritma KNN memiliki banyak rumus jarak. Salah satunya yaitu 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).

Pada percobaan Algoritma KNN diatas menggunakan Rumus Jarak Euclidean. Dimana terdapat 3 kali hasil percobaan untuk mengetahui kelas dari setiap nilai yang di inputkan pada program Data Testing. Dimana nilai tersebut nantinya digunakan untuk memberikan keputusan bahwa pengguna atau pelaku masuk pada ketegori kelas mana. Terdapat 3 kelas yakni “Beasiswa, Reguler, dan Tidak Lulus”. Nantinya ketika menginputkan data nilai pada program, kemudian program akan mengeksekusi dan akan menampilkan output yaitu berupa keputusan. Pada percobaan diatas didapatkan 3 output berbeda dengan nilai inputan yang berbeda pula. Percobaan pertama menghasilkan keputusan kelas Beasiswa. Percobaan kedua menghasilkan kelas Tidak Lulus, dan percobaan ketiga menghasilkan kelas Reguler.

VII. Referensi

- “Apa itu Machine Learning? Beserta pengertian dan Cara Kerjanya.”, 19 Agustus 2020. Diakses 20 Januari 2021. <https://www.dicoding.com/blog/machine-learning-adalah/>
- “Apa itu Machine learning?”, 14 Juli 2020. Diakses 20 Januari 2021. <https://www.selamatpagi.id/pengertian-machine-learning/>
- “Cara Kerja Algoritma k-Nearest Neighbor (k-NN)”. 17 Agustus 2018. Diakses 20 januari 2021. <https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e>
- Rivki, Muhammad.dkk. IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR DALAM PENGKLASIFIKASIAN FOLLOWER TWITTER YANG MENGGUNAKAN BAHASA INDONESIA. *Media.neliti.com*. Diakses 20 januari 2021.
- Mustakim. Giantika Oktaviani F. Algoritma *K-Nearest Neighbor Classification* Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa. *Ejournal.uin.suska.ac.id*. diakses 20 Januari 2021.
- Rauhan, Ahmad. Pengolahan Data Menggunakan *Machine Learning*. *Library.universitaspertamina.ac.id*. diakses 20 januari 2021.
- Wahyono, Teguh. Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan. www.researchgate.net. Diakses 20 januari 2021.
- Purba, Muhammad Ali. IMPLEMENTASI ALGORITMA NEAREST NEIGHBOR DALAM MENGUKUR TINGKAT KEPUASAN MASYARAKAT PADA PELAYANAN SAMSAT MEDAN SELATAN. *Ejurnal.stmik-budidarma.ac.id*. diakses 21 januari 2021.
- Aditya, Mohammad Adhisyanda.dkk. Penggabungan Teknologi Untuk Analisa Data Berbasis Data Science. *Seminar-id.com*. diakses 21 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.uii.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.
- Sinuhaji, Imelia Rosa Br.dkk. IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR UNTUK MENENTUKAN MAHASISWA BERPRESTASI DI STMIK KRISTEN NEUMANN INDONESIA. Media.neliti.com. Diakses 21 januari 2021.