LAPORAN PRAKTIKUM KECERDASAN BUATAN

LAPORAN PRAKTIKUM KECERDASAN BUATAN MECHINE LEARNING K-NEAREST NEIGHBORS DENGAN RUMUS JARAK EUCLIDEAN

Oleh

Aditya Roman Asyhari 361855401130 (3E)

PROGRAM STUDI DIPLOMA III
TEKNIK INFORMATIKA
POLITEKNIK NEGERI
BANYUWANGI 2020

1. **Tujuan Praktikum**

   Implementasi *K-Nearest Neighbors* (KNN) menggunakan java.

2. **Teori Dasar**

   a. **Mechine Learning dan Supervised Learning**

   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. 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.

   Teknologi *machine learning* (ML) adalah mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya. 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. 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.

   Machine Learning merupakan salah satu cabang dari disiplin ilmu Kecerdasan Buatan (*Artificial Intellegence*) yang membahas mengenai pembangunan sistem yang berdasarkan pada data. Salah satu teknik pengaplikasian *machine learning* adalah *supervised learning.* Seperti yang dibahas sebelumnya, *machine learning* tanpa data maka tidak akan bisa bekerja. Oleh karena itu hal yang pertama kali disiapkan adalah **data**. Data biasanya akan dibagi menjadi 2 kelompok, yaitu **data training** dan **data testing**. *Data training* nantinya akan digunakan untuk melatih algoritma untuk mencari model yang cocok, sementara data *testing* akan dipakai untuk

mengetes dan mengetahui performa model yang didapatkan pada tahapan *testing.* Dari model yang didapatkan, kita dapat melakukan prediksi yang dibedakan menjadi dua macam, tergantung tipe keluarannya. Jika hasil prediksi bersifat diskrit, maka dinamakan proses **klasifikasi**. Contohnya klasifikasi jenis kelamin dilihat dari tulisan tangan (output laki dan perempuan). Sementara jika kelurannya bersifat kontinyu, maka dinamakan proses **regresi**. Contohnya prediksi kisaran harga rumah di kota Bandung (output berupa harga rumah).

b. Algortima KNN

Algoritma *K-Nearest Neighbor* (KNN**)** adalah sebuah metode untuk melakukan klasifikasi terhadap obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut. Prinsip kerja dari *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (neighbor) terdekatnya dalam data pelatihan.

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah *k* buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Nilai *k* yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai *k* yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *k-nearest neighbor*.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

c. Rumus Jarak KNN

1. **Euclidean**

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

2. **Manhattan**

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

3. **Minkowsky**

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

4. **Chebychev**

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

5. **Hamming**

   Jarak Hamming adalah cara mencari jarak antar 2 titik yang dihitung dengan panjang vektor biner yang dibentuk oleh dua titik tersebut dalam block kode biner.

6. **Source Code**

   a. **Kelas java DataLatih**

```
package KNN;


/**
 *
 * @author root
 */
public class DataLatih {
   // Data Latih
   int[] NR = {5, 4, 9, 6, 10, 9, 3, 5, 9, 2};
```

```
    int[] UN = {6, 6, 8, 5, 8, 6, 4, 3, 4, 3};
    String[] kelas = {"REGULER", "REGULER", "BEASISWA",
"REGULER", "BEASISWA",
                    "BEASISWA", "TIDAK LULUS", "TIDAK LULUS",
"REGULER", "TIDAK LULUS"};
}
```

b. Kelas java Data Uji

```
package KNN;

import java.io.IOException;

/**
 *
 * @author root
 */
public class KNN_run {
    public static void main(String[] args) throws IOException {
        // TODO code application logic here
        //Data Testing

        int nr = 6;
        int un = 9;

        KNN_algoritma deteksi = new KNN_algoritma();
        String hasil = null;

        hasil = deteksi.knn(nr, un);
    }
}
```

c. Kelas java Algoritma KNN

```
package KNN;
```

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

/**
 *
 * @author root
 */
public class KNN_algoritma {
    public static String ranking = "";
    public String knn(int nR, int nU) throws IOException{
        //Data Training
        DataLatih latih = new DataLatih();

        //Data Testing
        int NR = nR;
        int UN = nU;

        double hasilED, hasilED_before;

        Map knn = new HashMap<>();
        System.out.println("DATA TESTING");
        System.out.println("NR : " + NR);
        System.out.println("UN : " + UN);
        System.out.println("");
        System.out.println("Hasil Jarak Euclidean");
        System.out.println("Nilai Rapot \t" + "Nilai UN\t" + "Kelas \t" +
"\t ED");
```

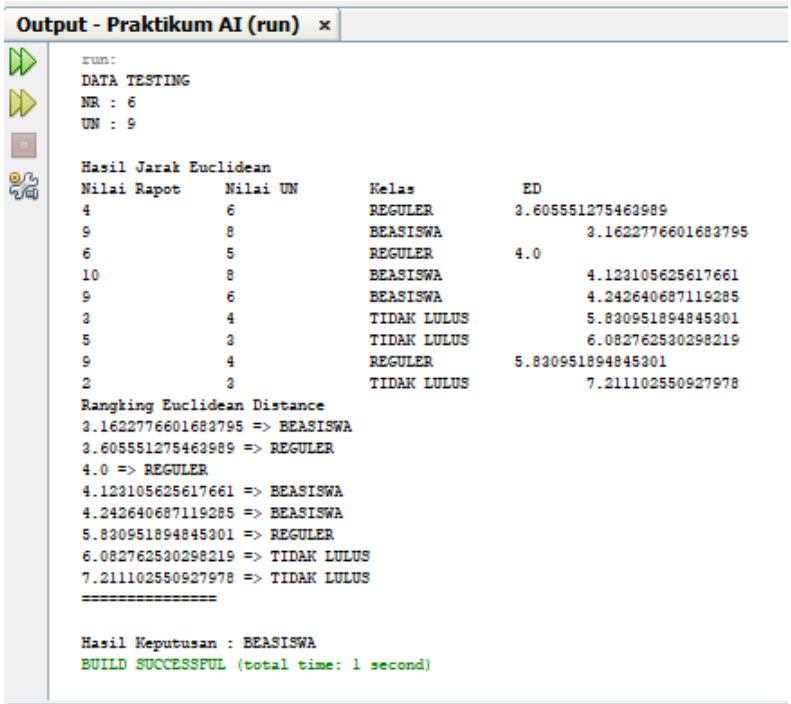
```
        for (int a = 1; a<10; a++){
            hasilED = jarakEuclidean(latih.NR[a], NR, latih.UN[a], UN);
            System.out.println(latih.NR[a] + "\t \t" + latih.UN[a] + "\t \t" +
latih.kelas[a] + "\t \t" + hasilED);
            knn.put(hasilED, latih.kelas[a]);
        }
        System.out.println("Rangking Euclidean Distance");
        Map knn_hasil = new TreeMap(knn);
        printMap(knn_hasil);

        System.out.println("");
        System.out.println("Hasil Keputusan : " +
knn_hasil.values().toArray()[0]);
        return (String) knn_hasil.values().toArray()[0];
    }

    public static void printMap(Map map){
        Set s = map.entrySet();
        Iterator it = s.iterator();
        int i = 0;
        while (it.hasNext()){
            Map.Entry entry = (Map.Entry) it.next();
            Double key = (double) entry.getKey();
            String value = (String) entry.getValue();
            ranking = ranking + "\n" + "Apel: " + i + "nilai: " + key + "=>"
+ value;
            System.out.println(key + " => " + value);
            i++;
        }
        System.out.println("===============");
    }
    // Membuat function untuk jarak Euclidean
```

```
    static double jarakEuclidean(int R1, int R2, int G1, int G2){
        return Math.sqrt(Math.pow(R1 - R2, 2) + Math.pow(G1 - G2, 2));
    }
}
```

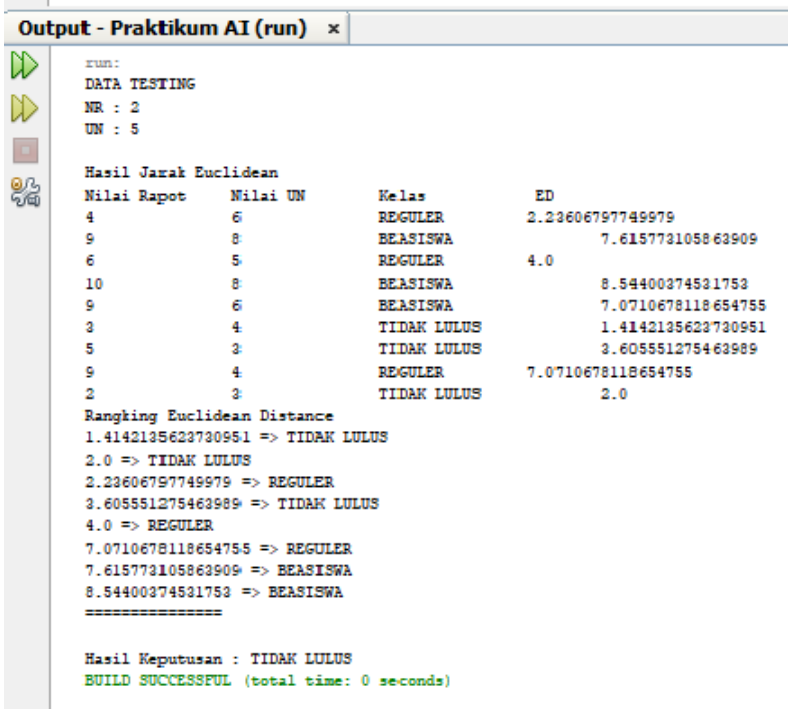
7. Hasil



```
Output - Praktikum AI (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
Rangking 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: 1 second)
```



```
Output - Praktikum AI (run)
run:
DATA TESTING
NR : 2
UN : 5

Hasil Jarak Euclidean
Nilai Rapot     Nilai UN        Kelas           ED
4               6               REGULER         2.23606797749979
9               8               BEASISWA                7.615773105863909
6               5               REGULER         4.0
10              8               BEASISWA                8.54400374531753
9               6               BEASISWA                7.071067811865475
3               4               TIDAK LULUS             1.4142135623730951
5               3               TIDAK LULUS             3.605551275463989
9               4               REGULER         7.071067811865475
2               3               TIDAK LULUS             2.0
Rangking Euclidean Distance
1.4142135623730951 => TIDAK LULUS
2.0 => TIDAK LULUS
2.23606797749979 => REGULER
3.605551275463989 => TIDAK LULUS
4.0 => REGULER
7.071067811865475 => REGULER
7.615773105863909 => BEASISWA
8.54400374531753 => BEASISWA
===============

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

8. Kesimpulan

Algoritman KNN merupakan algoritma yang bisa melakukan prediksi. Cara yang digunakan sangat sederhana, cukup menghitung jarak terdekat. Artinya, apabila ada input objek baru yang tidak dikenali, algoritma KNN akan mencari objek terdekat dengan objek yang baru diinput(dalam database). Cara menghitung jarak terdekatnya adalah dengan menggunakan berbagai macam rumus yang sudah diuraiakan diatas. Algoritma ini adalah algoritma yang paling simple dari semua algoritma mechinr learning, dan bisa dipakai untuk mesin pencari.

9. Referensi

a. https://www.codepolitan.com/mengenal-teknologi-machine-learning-pembelajaran-mesin
b. https://www.dicoding.com/blog/machine-learning-adalah/
c. https://E-journal.uajy.ac.id/2424/3/2TF04773.pdf
d. https://belajarkalkulus.com/clustering-part-iii/