

Silahkan cari satu tutorial yang membahas tentang pengukuran jarak antar data. Buat ringkasan dari tutorial tersebut dan tulis dalam format ms word, lalu kumpulkan sebelum batas waktu. Sertakan sumber link tutorial yang diringkas.

Terima kasih.

**NAMA : MEIGI RAHMAN**

**NIM : 202420049**

## **Tugas 01**

Silahkan cari satu tutorial yang membahas tentang pengukuran jarak antar data. Buat ringkasan dari tutorial tersebut dan tulis dalam format ms word, lalu kumpulkan sebelum batas waktu. Sertakan sumber link tutorial yang diringkas.

### **Jarak Antar Objek**

Semakin mirip suatu objek dengan objek lainnya, maka letak objek tersebut akan semakin dekat. Semakin berbeda suatu objek dengan objek lainnya, maka letak objek tersebut akan semakin jauh.

Pengukuran kedekatan antar objek bergantung dari beberapa faktor seperti skala data (nominal, ordinal, interval, rasio). Hubungan antar variabel juga turut mempengaruhi pemilihan jarak yang digunakan.

#### **1. Jarak Untuk Atribut Nominal**

Atribut nominal mempunyai k kategori yang tidak memiliki urutan yang berarti

#### **2. Jarak Untuk Atribut Ordinal**

Atribut ordinal mempunyai ciri yaitu kategori-nya memiliki urutan yang berarti.

#### **3. Jarak Untuk Atribut Numerik**

Atribut numerik mempunyai ciri dapat dilakukan operasi penjumlahan dan/atau pembagian. Atribut ini terdiri dari atribut interval dan rasio. Beberapa perhitungan jarak yang bisa digunakan untuk atribut numerik adalah

- 1) Euclidian distance
- 2) Manhattan distance
- 3) Minkowski distance
- 4) Supreme distance (Chebyshev distance)
- 5) Mahalanobis distance

#### **4. Jarak Untuk Atribut Campuran**

Pada prakteknya, variabel yang ada tidak hanya berupa atribut nominal saja, atau ordinal saja, atau numerik saja. Seringkali kita dihadapkan pada kasus dengan lebih dari 1 atribut. Perhitungan jarak berbeda bergantung dari jenis atribut. Jika atribut nominal, maka jarak yang digunakan jarak untuk atribut nominal.

### **Mengukur Jarak Euclidean: Teori dan Implementasi Menggunakan Java**

Jarak Euclidean biasa diterapkan untuk membantu proses klasifikasi pada data mining. Contoh algoritma yang menggunakan jarak ini adalah k-nearest neighbor. Jarak euclidean juga bisa diterapkan untuk algoritma image retrieval (temu kembali citra).

**NAMA : MEIGI RAHMAN**  
**NIM : 202420049**

Euclidean ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi.

Contoh jarak euclidean 1 dimensi

Misalnya, terdapat titik  $x_1$  bernilai 10 dan  $x_2$  bernilai -5. Maka kurangkan  $x_1 - x_2 = 10 - (-5) = 15$ . Lalu 15 kuadratkan sehingga memperoleh hasil 225. Langkah terakhir adalah diakarkan 2, sebagaimana rumusnya, sehingga kembali ke hasil 15. Bagaimana jika prosesnya dibalik, bukan  $x_1 - x_2$  tapi  $x_2 - x_1$ ? hasilnya akan sama saja. Coba saja!

Contoh jarak euclidean 2 dimensi

Terdapat suatu studi kasus begini, saya memiliki teman bernama anton yang berusia 25 tahun dengan berat badan 65 kg. Teman saya satu lagi, andi berusia 27 tahun dengan berat badan 70 kg. Sedangkan saya sendiri berumur 26 tahun dengan berat badan 67 kg. Pertanyaannya, berdasarkan variabel usia dan berat badan, saya mirip anton atau andi?

Pertanyaan itu bisa dijawab menggunakan pendekatan jarak euclidean. Anggap saja variabel usia adalah  $x$  dan variabel berat badan adalah  $y$ . Maka, implementasinya seperti ini:

$d(\text{anton, saya}) = (25-26)^2 + (65-67)^2 = 1+4 = 5$ . Hasil dari akar 5 adalah 2,24

$d(\text{andi, saya}) = (27-26)^2 + (70-67)^2 = 1+9 = 10$ . Hasil dari akar 10 adalah 3,16

semakin kecil nilai jarak euclidean maka semakin mirip

Contoh penerapan jarak euclidean di Java

```
public class jarakEuclidean {  
  
    static double jarakEuclidean(int x1, int x2, int y1, int y2) {  
        return Math.sqrt(Math.pow(x1 - x2, 2) + Math.pow(y1 - y2, 2));  
    }  
  
    public static void main(String[] args) {  
  
        int umurSaya = 26 ;  
        int BBSaya=67;  
  
        int umurAnton = 25;  
        int BBAnton = 65;  
  
        int umurAndi = 27;  
        int BBAndi = 70;  
  
        double d_saya_anton= jarakEuclidean(umurAnton,umurSaya,BBAnton,BBSaya);  
        double d_saya_andi= jarakEuclidean(umurAndi,umurSaya,BBAndi,BBSaya);  
  
        System.out.println("jarak antara saya dengan anton = " + d_saya_anton);  
        System.out.println("jarak antara saya dengan andi = " + d_saya_andi);  
  
    }  
}
```

**NAMA : MEIGI RAHMAN**  
**NIM : 202420049**

Untuk code akar dan pangkat, bisa menggunakan objek Math. Pilih Math.sqrt untuk akar dan Math.pow untuk pangkat.

Hasil running-nya

```
run:  
jarak antara saya dengan anton = 2.23606797749979  
jarak antara saya dengan andi = 3.1622776601683795  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 3. Running jarak euclidean

Sumber :

Maulana, Arofiqi. 2021. Jarak Antar Objek. <https://arofiqimaulana.com/konsep-jarak/> (diakses tanggal 23 Maret 2021)

Fauzan, Charis. 2019. Mengukur Jarak Euclidean: Teori dan Implementasi Menggunakan Java. <https://www.kitainformatika.com/2019/10/mengukur-jarak-euclidean-teori-dan.html> (diakses tanggal 23 Maret 2021)

Nama : Mohammad Ilham  
Nim : 202420046  
Mata Kuliah : Advanced Database

## TUGAS 1

Silahkan cari satu tutorial yang membahas tentang pengukuran jarak antar data. Buat ringkasan dari tutorial tersebut dan tulis dalam format ms word, lalu kumpulkan sebelum batas waktu. Sertakan sumber link tutorial yang diringkas.

### Jawaban

Tutorial ini membahas mengenai pengelompokan data peserta didik di sekolah untuk memprediksi calon penerima beasiswa. Ada beberapa cara yang dapat dilakukan untuk menghitung jarak data yaitu antara lain dengan menggunakan *Euclidean distance*, *Manhattan distance*, *Chebisev distance*.

Disini saya menggunakan ***Euclidean distance***. Metode yang umumnya digunakan dalam pengukuran jarak antara data point salah satunya adalah *Euclidean Distance*. *Euclidean distance* merupakan metode yang dikembangkan berdasarkan teorema pythagoras.

#### A. Pengukuran Jarak Antar Data

Dalam algoritma k-means banyaknya cluster ditentukan terlebih dahulu, dengan syarat banyaknya cluster harus lebih kecil dari banyaknya data ( $k < n$ ). Banyaknya cluster pada aplikasi data mining ini adalah 2 ( $k=2$ ). Selanjutnya menentukan centroid awal dengan metode mengambil data dari data point secara acak. Centroid merupakan titik tengah suatu cluster yang berupa nilai, dalam algoritma k-means untuk menentukan centroid awal ditentukan dengan mengambil data point secara acak (Tabel 1).

Sebanyak lima data point diambil sebagai sampel dengan nilai dan gaji orang tua yang dikelompokkan diidentifikasi sebagai kelompok nilai dan kelompok gaji. Digunakan kelompok nilai dan kelompok gaji agar memudahkan dalam perhitungan jarak data dengan centroid dengan menggunakan rumus euclidean distance. Kelompok nilai dan kelompok gaji harus sama jumlahnya, dan untuk memudahkan perhitungan diberikan kelompok nilai dan kelompok gaji sebanyak 1 - 4 kelompok yang masing-masing dibandingkan (Tabel 2).

C1 merupakan centroid pada cluster pertama, dan C2 merupakan centroid pada cluster dua yang ditentukan secara acak. Setelah centroid awal ditentukan, menghitung jarak data dengan centroid menggunakan rumus euclidean distance. Menghitung jarak data dengan centroid awal :

$$d(x_1, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_2, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_3, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_4, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_5, c_1) = \sqrt{(3 - 3)^2 + (2 - 1)^2} = 1$$

$$d(x_1, c_2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x_2, c_2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x_3, c_2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x_4, c_2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x_5, c_2) = \sqrt{(3 - 3)^2 + (2 - 2)^2} = 0$$

Tabel 1. Data Point

No	Kel. Nilai	Kel. Gaji
1	3	1
2	3	1
3	3	1
4	3	1
5	3	2

Tabel 2. Centroid Awal

No	Kel. Nilai	Kel. Gaji
C1	3	1
C2	3	2

Tabel 3. Hasil Perhitungan Jarak Awal

No	Kel. Nilai	Kel. Gaji	DC1	DC2	C1	C2
1	3	1	0	1	V	
2	3	1	0	1	V	
3	3	1	0	1	V	
4	3	1	0	1	V	
5	3	2	1	0		v

Tabel 4. Centroid Pengulangan ke-I

No	Kel. Nilai	Kel. Gaji
C1	3	1
C2	3	1

Tabel 5. Hasil Perhitungan Jarak Pada Pengulangan ke-I

No	Kel. Nilai	Kel. Gaji	DC1	DC2	C1	C2
1	3	1	0	1	V	
2	3	1	0	1	V	
3	3	1	0	1	V	
4	3	1	0	1	V	
5	3	2	1	0		v

Hasil perhitungan jarak antara datapoint dengan centroid awal dengan menggunakan rumus euclidean distance, dan untuk mencari cluster pertama atau dua dapat dilihat pada data yang memiliki nilai terkecil, hasil perhitungan jarak didefinisikan sebagai DC1 dan DC2 (Tabel 3).

Hasil perhitungan jarak data point dengan centroid awal dihasilkan, jika  $DC1 < DC2$  maka masuk ke cluster pertama (C1). Namun jika  $DC2 < DC1$  maka masuk ke cluster dua (C2). Untuk centroid selanjutnya ditentukan berdasarkan data yang diambil dari rata-rata masing-masing cluster (Tabel 4).

Menghitung jarak data dengan centroid pada pengulangan ke-1 :

$$d(x_1, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_2, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_3, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_4, c_1) = \sqrt{(3 - 3)^2 + (1 - 1)^2} = 0$$

$$d(x_5, c_1) = \sqrt{(3 - 3)^2 + (2 - 1)^2} = 1$$

$$d(x1,c2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x2,c2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x3,c2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x4,c2) = \sqrt{(3 - 3)^2 + (1 - 2)^2} = 1$$

$$d(x5,c2) = \sqrt{(3 - 3)^2 + (2 - 2)^2} = 0$$

Sumber :

Wicaksono, A. E. (2016). IMPLEMENTASI DATA MINING DALAM PENGELOMPOKAN DATA. *Jurnal Teknik Informatika*, 206-216.

Link : <https://www.youtube.com/watch?v=tYyCUnF1yd8>

Nama : Rike Sucihati

Salah satu cara menghitung jarak antar dua instance/data adalah dengan menggunakan Euclidean Distance yang didasarkan teori segitiga pitagoras. Euclidean Distance akan menghasilkan nilai jarak antara dua objek. Nilai jarak bisa dimulai dari 0 sampai dengan tak terhingga. Dengan adanya jarak antar data, maka proses penentuan target kelas dapat ditentukan dengan bergantung dengan jumlah K.

Jarak Euclidean biasa diterapkan untuk membantu proses klasifikasi pada data mining. Contoh algoritma yang menggunakan jarak ini adalah k-nearest neighbor. Jarak euclidean juga bisa diterapkan untuk algoritma image retrieval.

### **Contoh jarak euclidean 1 dimensi**

Misalnya, terdapat titik  $x_1$  bernilai 10 dan  $x_2$  bernilai -5. Maka kurangkan  $x_1 - x_2 = 10 - (-5) = 15$ . Lalu 15 kuadratkan sehingga memperoleh hasil 225.

Langkah terakhir adalah diakarkan 2, sebagaimana rumusnya, sehingga kembali ke hasil 15. Bagaimana jika prosesnya dibalik, bukan  $x_1 - x_2$  tapi  $x_2 - x_1$ ? hasilnya akan sama saja.

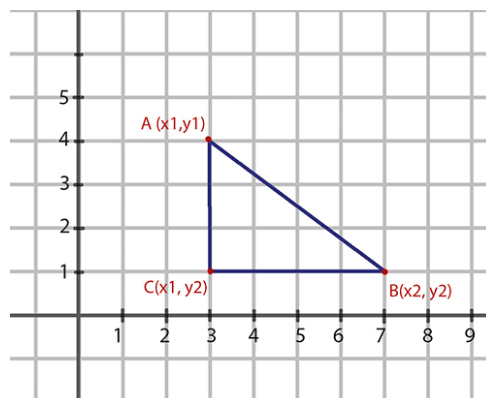


Nama : Yuli Apriyanti

Nim : 222420051

# Mengukur Jarak Euclidean: Teori dan Implementasi Menggunakan Java

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Artikel ini menjelaskan teori singkat beserta penerapannya di pemrograman java.



Gambar 1. Ilustrasi Pengukuran Jarak Euclidean

Jarak Euclidean biasa diterapkan untuk membantu proses klasifikasi pada data mining. Contoh algoritma yang menggunakan jarak ini adalah k-nearest neighbor. Jarak euclidean juga bisa diterapkan untuk algoritma image retrieval (temu kembali citra). Maka, mengenai penerapan jarak euclidean untuk berbagai algoritma sudah saya jabarkan di artikel lain.

Jarak euclidean diterapkan untuk berbagai dimensi. Agar memudahkan pemahaman, saya sertakan contoh jarak euclidean untuk 1 dimensi dan 2 dimensi. Secara umum, rumus jarak euclidean ditunjukkan oleh Gambar 2. Gambar 2 menunjukkan rumus jarak euclidean 2 dimensi karena melibatkan variabel x dan y.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Gambar 2. Rumus Jarak Euclidean

### Contoh jarak euclidean 1 dimensi

Misalnya, terdapat titik  $x_1$  bernilai 10 dan  $x_2$  bernilai -5. Maka kurangkan  $x_1 - x_2 = 10 - (-5) = 15$ . Lalu 15 kuadratkan sehingga memperoleh hasil 225. Langkah terakhir adalah diakarkan 2, sebagaimana rumusnya, sehingga kembali ke hasil 15. Bagaimana jika prosesnya dibalik, bukan  $x_1 - x_2$  tapi  $x_2 - x_1$ ? hasilnya akan sama saja. Coba saja.

### Contoh jarak euclidean 2 dimensi

Terdapat suatu studi kasus begini, saya memiliki teman bernama anton yang berusia 25 tahun dengan berat badan 65 kg. Teman saya satu lagi, andi berusia 27 tahun dengan berat badan 70 kg. Sedangkan saya sendiri berumur 26 tahun dengan berat badan 67 kg. Pertanyaannya, berdasarkan variabel usia dan berat badan, saya mirip anton atau andi?

Pertanyaan itu bisa dijawab menggunakan pendekatan jarak euclidean. Anggap saja variabel usia adalah  $x$  dan variabel berat badan adalah  $y$ . Maka, implementasinya seperti ini:

$$d(\text{anton, saya}) = (25-26)^2 + (65-67)^2 = 1+4 = 5. \text{ Hasil dari akar 5 adalah } 2,24$$

$$d(\text{andi, saya}) = (27-26)^2 + (70-67)^2 = 1+9 = 10. \text{ Hasil dari akar 10 adalah } 3,16$$

Kesimpulannya, saya mirip siapa? anda sudah tahu jawabannya. Mana jarak yang paling dekat diantara 2,24 dan 3,16. Tentu saja 2,24. Jadi, semakin kecil nilai jarak euclidean maka semakin mirip. Jika hasilnya 0? berarti bukan hanya mirip tapi sama.

Sekarang, bagaimana cara menerapkan jarak euclidean di Java? Mudah saja. Buat method baru bernama jarakEuclidean, tuliskan rumus euclidean di method itu. Untuk code akar dan pangkat, bisa menggunakan objek Math. Pilih Math.sqrt untuk akar dan Math.pow untuk pangkat. Parameter pada method ini ada 4, ini khusus untuk 2 dimensi, yakni  $x_1, x_2, y_1, y_2$ . Jika 3 dimensi maka ada 6 parameter. Tinggal anda sesuaikan saja. Selanjutnya panggil method jarakEuclidean pada method main. Oke beres, seperti ini:

```
public class jarakEuclidean {
    static double jarakEuclidean(int x1, int x2, int y1, int y2) {
```

```

        return Math.sqrt(Math.pow(x1 - x2, 2) + Math.pow(y1 - y2, 2));
    }
    public static void main(String[] args) {
        int umurSaya = 26 ;
        int BBSaya=67;

        int umurAnton = 25;
        int BBAnton = 65;

        int umurAndi = 27;
        int BBAndi = 70;

        double d_saya_anton= jarakEuclidean(umurAnton,umurSaya,BBAnton,BBSaya);
        double d_saya_andi= jarakEuclidean(umurAndi,umurSaya,BBAndi,BBSaya);

        System.out.println("jarak antara saya dengan anton = " + d_saya_anton);
        System.out.println("jarak antara saya dengan andi = " + d_saya_andi);

    }
}
}

```

Hasil running-nya seperti Gambar 3:

```

run:
jarak antara saya dengan anton = 2.23606797749979
jarak antara saya dengan andi = 3.1622776601683795
BUILD SUCCESSFUL (total time: 2 seconds)

```

Gambar 3. Running jarak Euclidean

ALAMAT LINK : <https://www.youtube.com/watch?v=tYyCUnF1yd8>

**Nama : Achmad Nopransyah**

**NIM : 202420050**

**Program Studi : Magister Teknik Informatika**

### **Soal**

Silahkan cari satu tutorial yang membahas tentang pengukuran jarak antar data. Buat ringkasan dari tutorial tersebut dan tulis dalam format ms word, lalu kumpulkan sebelum batas waktu. Sertakan sumber link tutorial yang diringkas.

### **Jawab :**

**Dalam hal ini yang akan diringkas ialah pengukuran jarak antar data dengan metode Euclidean,**

**Link Tutorial : [https://www.youtube.com/watch?v=x\\_pJXbHNICU&list=PLJ-svfUNrVUbTeL8MRfUvRTtaCXbpS3UJ&index=16](https://www.youtube.com/watch?v=x_pJXbHNICU&list=PLJ-svfUNrVUbTeL8MRfUvRTtaCXbpS3UJ&index=16)**

**Judul : Ilmu Data #10 - Metrik Jarak: Euclidean**

Adapun dalam tutorial Pengukuran jarak antar data tersebut dijabarkan mengenai berbagai hal diantaranya :

1. Diawal dijelaskan mengenai Koordinat Cartesian yang digunakan untuk memplot data point yang terdiri atas x dan y, dari sini dapat dijelaskan mengenai pengertian umum dari metode Euclidean tersebut yaitu Euclidean distance yang merupakan metode perhitungan jarak dari 2 buah titik dalam Euclidean space (contohnya koordinat cartesian).
2. Selanjutnya dijelaskan bahwa Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E.

3. Selanjutnya dijelaskan mengenai teknik pengukuran Euclidean pada 1 dimensi yang didapatkan rumus

$$d(a,c)=|A-C|$$

4. Selanjutnya dijelaskan teknik pengukuran Euclidean pada 2 dimensi, pada tutorial diberikan contoh terdapat 4 kotak yang memiliki tinggi dan panjang selanjutnya dicari Euclidean Distance dari titik a ke titik c, dari sini dapat diketahui bahwa Euclidean ini berkaitan dengan Teorema Pythagoras dan biasanya, sehingga didapatkan rumus

$$d(a,c) = \sqrt{(H_c - H_a)^2 + (L_c - L_a)^2}$$

5. Selanjutnya pada tutorial dijelaskan mengenai teknik pengukuran Euclidean pada 3 dimensi

$$d(a,c) = \sqrt{(H_c - H_a)^2 + (L_c - L_a)^2 + (W_c - W_a)^2}$$

6. Sehingga apabila diterapkan pada n variable (n dimensi) maka didapatkan rumus

$$a=(a_1, a_2, a_3, \dots, a_n) \quad b=(b_1, b_2, b_3, \dots, b_n)$$
$$d(a,b) = \sum_{i=1}^n (b_i - a_i)^2$$

Nama : Febra Oktariansyah  
Nim : 202420047  
Mata Kuliah : Advanced Database

Berikut cara atau tahapan dalam pengukuran jarak antar data dengan algoritma K-Means Clustering adalah sebagai berikut :

1. Tentukan jumlah cluster lalu menginput data yang digunakan untuk menentukan nilai rata-rata data dalam satu cluster dan menentukan jarak dari setiap data ke centroid.
2. Alokasikan data ke dalam cluster secara random atau acak
3. Hitung centroid/rata-rata dari data yang ada di masing-masing cluster, nilai centroid pada k-means digunakan sebagai pusat cluster. Dengan menentukan anggota cluster secara acak pada tahap sebelumnya, maka terbentuk iterasi awal sebagai pusat cluster acak.
4. Alokasikan masing-masing data ke centroid/rata-rata terdekat, pada tahap ini hasil centroid dari setiap cluster sudah diketahui, kemudian data dialokasikan pada centroid terdekat berdasarkan nilai jarak similarity data terhadap centroid. Jarak similarity dari data ke centroid pada masing-masing cluster diperoleh dari perhitungan Euclidean Distance. Kemudian nilai jarak setiap data ke centroid cluster dibandingkan, dan data point menjadi anggota dari cluster berdasarkan jarak data ke centroid terdekat.
5. Konvergen, mengalokasikan data point ke centroid dengan nilai jarak terdekat, dengan menguji apakah cluster yang terbentuk telah membentuk cluster yang konvergen atau tidak. Cluster dinyatakan konvergen jika anggota dari masing-masing cluster yang terbentuk tidak mengalami perubahan anggota. Tetapi jika masih terjadi perubahan anggota cluster, maka akan kembali dilakukan tahapan menghitung centroid dari masing-masing cluster yang terbentuk dan diikuti dengan perhitungan nilai similarity ke centroid yang baru terbentuk. Proses tersebut terus berulang sampai hasil cluster konvergen.

Sumber : <https://informatikalogi.com/algoritma-k-means-clustering/>  
<https://ejournal.gunadarma.ac.id/index.php/tekno/article/view/1599/1358>  
[https://www.youtube.com/watch?v=8mXL2z3lg\\_o&t=11s](https://www.youtube.com/watch?v=8mXL2z3lg_o&t=11s)

## TUGAS I

Kemiripan (*similarity*) adalah ukuran numerik dimana dua objeknya mirip, nilai 0 jika tidak mirip dan nilai 1 jika mirip penuh. Sementara ketidakmiripan (*dissimilarity*) adalah derajat numerik dimana dua objek yang berbeda, jangkauan nilai 0 sampai 1 atau bahkan sampai  $\infty$ .

### I. Ketidakmiripan dan Kemiripan Data Satu Atribut

Istilah ketidakmiripan juga dapat disebut sebagai ukuran jarak (*distance*) antara dua data. Jika  $s$  adalah ukuran kemiripan dan  $d$  adalah ukuran ketidakmiripan, serta jika interval/range nilainya adalah  $[0,1]$ , maka dapat dirumuskan bahwa  $s+d=1$ . Sebenarnya ukuran kemiripan dan ketidakmiripan tidak harus selalu dalam interval  $[0,1]$ , tetapi boleh juga menggunakan interval seperti  $[0,10]$ ,  $[0,100]$ , bahkan menggunakan nilai negative seperti  $[-1,1]$ ,  $[-10,10]$  dan sebagainya. Transformasi nilai  $s$  dan  $d$  tidak hanya terbatas pada formula  $s+d=1$ , karena ada juga yang menggunakan  $s = 1 / 1+d$  atau  $s = e^{-d}$ .

Pada metode tertentu dalam klasifikasi, ada juga yang mengharuskan agar nilai interval ketidakmiripan data harus ditransformasi dalam interval yang ternormalisasi  $[0,1]$ . Sebagai contoh, ada data dengan nilai ketidakmiripan  $\{10, 12, 25, 30, 40\}$  dengan intervalnya  $[10,40]$ . Jika akan ditransformasi ke dalam interval  $[0,1]$ , kita bisa menggunakan formula  $x = (x - \min(x)) / (\max(x) - \min(x))$  sehingga nilai-nilai ketidakmiripan tersebut ditransformasi menjadi  $\{0, 0.667, 0.5, 0.6667, 1\}$ .

Untuk fitur yang menggunakan tipe ordinal, misalnya sebuah atribut yang mengukur kualitas produk dengan skala  $\{\text{rusak, jelek, sedang, bagus, sempurna}\}$ , skala tersebut harus ditransformasikan ke dalam nilai numerik, misalnya  $\{\text{rusak}=0, \text{jelek}=1, \text{sedang}=2, \text{bagus}=3, \text{sempurna}=4\}$ . Kemudian, ada dua produk P1 dengan kualitas bagus dan P2 dengan kualitas jelek. Jarak (ketidakmiripan) antara P1 dan P2 dapat dihitung dengan cara  $D(P1,P2) = 3-1 = 2$ , atau jika dalam interval  $[0,1]$  menjadi  $3-1/4 = 0.5$ , sedangkan nilai kemiripannya adalah  $1-0.5 = 0.5$ .

Untuk atribut bertipe numerik (interval dan rasio), nilai ketidakmiripan didapat dari selisih absolut di antara dua data. Misalnya atribut usia, jika P1 adalah usia 45 dan P2 usia 25, sedangkan jangkauan nilai usia dalam data adalah  $[5,75]$ , nilai ketidakmiripan P1 dan P2 adalah  $D(P1,P2) = 45-25 = 20$ , atau jika dalam interval  $[0,1]$  menjadi  $20-5/75-5 = 0.21$ , sedangkan nilai kemiripannya adalah  $1-0.21 = 0.79$ .

**Tabel 1.** Formula Kemiripan dan Ketidakmiripan dengan Satu Atribut

Typ e Atribut	Kemiripan	Ketidakmiripan
Nominal	$s = \begin{cases} 1 & \text{jika } x = y \\ 0 & \text{jika } x \neq y \end{cases}$	$d = \begin{cases} 0 & \text{jika } x = y \\ 1 & \text{jika } x \neq y \end{cases}$
Ordinal	$s = 1 - d$	$d =  x - y  / (n - 1)$
Interval dan Rasio	$s = -d, s = \frac{1}{1+d}, s = e^{-d},$ $s = \frac{d - \min(d)}{\max(d) - \min(d)}$	$D =  x - y $

## II. Ketidakmiripan Data Multiatribut

Terdapat banyak cara yang dapat digunakan untuk menghitung jarak (ketidakmiripan) dua objek dengan beberapa atribut, diantaranya:

### 1. Jarak Euclidian

Metode Euclidean membandingkan jarak minimum image pengujian (testing), dengan database image pelatihan (training). Jarak euclidean dari dua vektor  $x$  dan  $y$  dihitung dengan Persamaan 1:

$$D_{(x,y)} = \sqrt{\sum_{j=1}^n |x - y|^2}$$

Semakin kecil nilai  $d(x, y)$ , maka semakin mirip kedua vektor yang dicocokkan/dibandingkan. Sebaliknya semakin besar nilai  $d(x, y)$  maka semakin berbeda kedua vektor yang dicocokkan (Budi Santosa, 2007).

### 2. Jarak Manhattan/City Block

Merupakan salah satu pengukuran yang paling banyak digunakan meliputi penggantian perbedaan kuadrat dengan menjumlahkan perbedaan absolute dari variabelvariabel. Prosedur ini disebut blok absolute atau lebih dikenal dengan city block distance (lihat Persamaan 2).

$$D_{(x,y)} = \sum_{j=1}^n |x - y|$$

### 3. Jarak Chebyshev

$$D_{(x,y)} = \max_{j=1}^n (|x_j - y_j|)$$

Sebagai contoh kita akan melakukan pengukuran jarak antar objek dengan persamaan euclidean pada set data tabel 2 di bawah ini.

**Tabel 2.** Contoh Data Dua Dimensi

Point	x	y
P1	1	1
P2	4	1
P3	1	2



**Tabel 3.** Hasil Pengukuran Jarak Euclidean

<b>Euclidean</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>
<b>P1</b>	0	3	1
<b>P2</b>	3	0	3.16
<b>P3</b>	1	3.16	0

### III. Kemiripan Data Multiatribut

Terdapat beberapa cara yang dapat digunakan untuk menghitung kemiripan (similaritas) dua objek dengan beberapa atribut, diantaranya:

#### 1. Simple Matching (SMC) dan Jaccard Coefficients (J)

Metode yang dapat digunakan untuk menghitung kemiripan dua objek (vektor) biner. Misalkan kita akan menghitung kemiripan objek p dan q, dimana kedua objek tersebut terdiri dari beberapa atribut yang bernilai biner. Kemiripan antara dua objek tersebut dapat dihitung dengan kuantitas berikut:

M01 = Jumlah atribut dimana p adalah 0 dan q adalah 1

M10 = Jumlah atribut dimana p adalah 1 dan q adalah 0

M00 = Jumlah atribut dimana p adalah 0 dan q adalah 0

M11 = Jumlah atribut dimana p adalah 1 dan q adalah 1

Nilai dari M01, M10, M00 dan M11 dimasukan ke dalam persamaan Simple Matching (SMC) atau Jaccard Coefficients (J) di bawah ini:

SMC = number of matches / number of attributes

SMC =  $(M11 + M00) / (M01 + M10 + M11 + M00)$

J = number of 11 matches / number of not-both-zero attributes values

J =  $(M11) / (M01 + M10 + M11)$

#### 2. Cosine Similarity

Metode ini digunakan untuk menghitung dua objek (vektor) dokumen. Jika d1 dan d2 adalah dua vektor dokumen maka kemiripan antara dua vektor dokumen yang dihitung dengan cosine similarity adalah sebagai berikut:

$\text{Cos}(d1,d2) = (d1 \cdot d2) / (|d1| \cdot |d2|)$

#### 3. Extended Jaccard Coefficient (Tanimoto)

Merupakan pengembangan dari Jaccard Coefficient untuk menghitung kemiripan dua

objek (vektor) dengan atribut kontinu, dengan persamaan sebagai berikut:

$$T(p, q) = \frac{p \cdot q}{\|p\|^2 + \|q\|^2 - p \cdot q}$$

#### IV. Kemiripan Objek dengan Atribut Campuran

Terkadang ditemukan bahwa atribut (fitur) pada objek (vektor) terdiri dari tipe data campuran (tidak seragam). Tidak selalu semua atribut bertipe numerik (interval atau rasio) atau bertipe kategoris (nominal dan ordinal). Untuk menghitung nilai kemiripan objek dengan atribut campuran kita bisa gunakan persamaan berikut:

$$s(x, y) = \frac{\sum_{i=1}^r S_i(x, y)}{\sum_{i=1}^r w_i}$$

si merupakan ukuran kemiripan diantara fitur ke-I dari x dan y, sedangkan wi adalah faktor bobot yang berkorelasi dengan fitur ke-1. Cara menentukan faktor bobot adalah dengan ketentuan sebagai berikut:

$w_i = 0$  , apabila nilai fitur-I dari objek x dan y tidak teridentifikasi.

$w_i = 0$  , apabila nilai  $s_i = 0$ .

$w_i = 1$  , apabila selain dua syarat di atas.

Sumber Data :

<https://media.neliti.com/media/publications/233469-metode-nilai-jarak-guna-kesamaan-atau-ke-192fb9ef.pdf>

<https://repository.unikom.ac.id/49057/1/Pertemuan%204%20-%20Materi%20%5BDM%20-%202016%5D.pdf>

## **Tugas 01**

Google Maps merupakan aplikasi yang bisa diandalkan, karena dapat memudahkan pengguna dalam menentukan rute terbaik untuk menghindari kemacetan dan melihat jarak dari satu tempat ke tempat lainnya Berikut cara menggunakan aplikasi dari Ios, Android atau PC.

### **Cara Pertama di Iphone atau iPad:**

1. Buka aplikasi Google Maps
2. Tambah titik pertama
3. Tap lama dimana saja pada peta, maka akan muncul pin merah
4. Di bagian bawah tap nama tempat
5. Lalu gulir kebawah, pilih ukur jarak.

Cara Kedua:

1. Tambah satu beberapa titik
2. Pindah peta agar menjadi lingkaran hitam atau tanda x berada di titik berikutnya yang ingin ditambahkan
3. Tap tambahkan di bagian bawah
4. Tambahkan titik sampai selesai.

Cara Ketiga:

1. Di bagian bawah terlihat total jarak dalam mil atau km
2. Tap urungkan untuk menghapus titik terakhir. Lalu di kanan atas, tap lainnya untuk menghapus semua titik
3. Tap tanda panah di kiri atas untuk setelah selesai

### **Cara Pertama di Android:**

1. Buka Google Maps pada Android
2. Tap lama dimana saja pada peta, maka akan muncul pin merah
3. Di bagian bawah tap nama tempat
4. Lalu gulir kebawah, pilih ukur jarak.

Cara Kedua:

1. Tambah satu beberapa titik
2. Pindah peta agar menjadi lingkaran hitam atau tanda x berada di titik berikutnya yang ingin ditambahkan
3. Tap tambahkan di bagian bawah
4. Tambahkan titik sampai selesai.

**Cara Ketiga:**

1. Di bagian bawah terlihat total jarak dalam mil atau km
2. Tap urungkan untuk menghapus titik terakhir. Lalu di kanan atas, tap lainnya untuk menghapus semua titik
3. Tap tanda panah di kiri atas untuk setelah selesai

**Cara Mengukur Jarak di PC:**

1. Buka Google Maps
2. Klik kanan pada titik awal, pilih ukur jarak
3. Klik dimana saja pada peta untuk mengukur jarak yang diinginkan dan menambahkan titik
4. Seret atau klik titik untuk memindahkan atau menghapusnya
5. Di bagian bawah terlihat total jarak dalam mil atau km
6. Klik “Tutup” (X) setelah selesai di bagian bawah.

**Sumber:**

Samudro, A. (2019). Cara mengukur jarak di Google Maps Android, iOS, PC Diakses pada 27 Maret 2021, dari <https://tirto.id/cara-mengukur-jarak-di-google-maps-android-ios-dan-pc-eloh>