

# LAPORAN TUGAS PROGRAM 3 KECERDASAN BUATAN

## “k-Nearest Neighbor (kNN)”

Nama : Adhyfa Fahmy Hidayat

Kelas : IF-40-GAB01

### Deskripsi Masalah

Diberikan file DataTrain\_Tugas3\_AI.csv berupa himpunan data berisi 800 data yang memiliki 5 atribut input (X1, X2, X3, X4, dan X5) dan 1 output yang memiliki 4 kelas/label (0, 1, 2, dan 3). Bangunlah sebuah sistem klasifikasi menggunakan metode k-Nearest Neighbors untuk menentukan kelas/label data testing dalam file DataTest\_Tugas3\_AI.csv. Sistem membaca masukan file DataTrain\_Tugas3\_AI.csv dan DataTest\_Tugas3\_AI.csv dan mengeluarkan output berupa file TebakanTugas3.csv berupa satu kolom berisi 200 baris angka bernilai integer/bulat (0, 1, 2, atau 3) yang menyatakan kelas/label baris atau record yang bersesuaian pada file DataTest\_Tugas3\_AI.csv.

### Rancangan

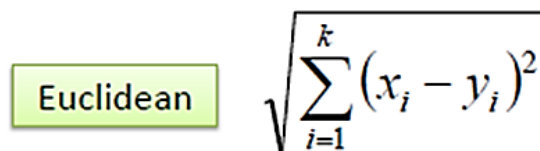
Pada kasus ini akan digunakan Bahasa pemrograman berupa python 3 untuk pengerjaannya. Untuk detail lebih lengkapnya dapat dilihat dari hasil kodingan

### Input

Terdapat dua fungsi yaitu load() dan load2(). Dimana load() merupakan inputan untuk memasukkan DataTrain\_Tugas3\_AI.csv yang akan displit menjadi trainingset dan testset. Sedangkan load2() merupakan inputan untuk memasukkan DataTrain\_Tugas3\_AI.csv sebagai trainingset dan DataTest\_Tugas3\_AI.csv sebagai testset

### Menghitung Euclidean Distance

Algoritma metode KNN sangatlah sederhana, bekerja dengan berdasarkan pada jarak terpendek dari sample uji ke sample latih untuk menentukan KNN nya. Setelah mengumpulkan KNN, kemudian diambil mayoritas dari KNN untuk dijadikan prediksi dari sample uji. Cara untuk menentukan jarak terpendek tersebut yaitu menggunakan rumus Euclidian distance.


$$\text{Euclidean} \quad \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Gambar 1 Rumus Euclidean

```
def euclidean(instance1, instance2, length):  
    distance = 0  
    for x in range(length):  
        distance += pow(instance1[x] - instance2[x], 2)  
    return math.sqrt(distance)
```

Gambar 2 Fungsi Euclidean

## Mencari Neighbor

Mencari k yang paling mirip. Nilai K dicari dengan cara mencoba-coba. Untuk membandingkan dan mengelompokkan hasil dari Euclidean yang telah dihitung sebelumnya. Lalu hasil terbanyak diambil menjadi kesimpulan. Pada kasus ini ditemukan k yang paling optimal yaitu k=3. Dengan tingkat akurasi yang dapat dilihat pada subbab selanjutnya

## Mencari Hasil Respon dari Dataset

Menghasilkan respons dari sekumpulan contoh data

## Menghitung Akurasi

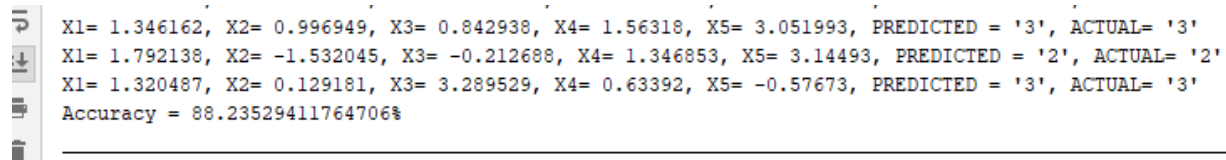
Menghitung tingkat akurasi dari prediksi yang dihasilkan dengan cara menghitung data yang benar dibagi jumlah data dikali seratus:

```
def getAccuracy(testset, predictions):
    correct = 0
    for x in range(len(testset)):
        if predictions[x] == testset[x][-1]:
            correct += 1
    akurasi = (correct/float(len(testset))) * 100
    return akurasi
```

Gambar 3 Fungsi Akurasi

## Menyatukan Semuanya dalam satu Fungsi main()

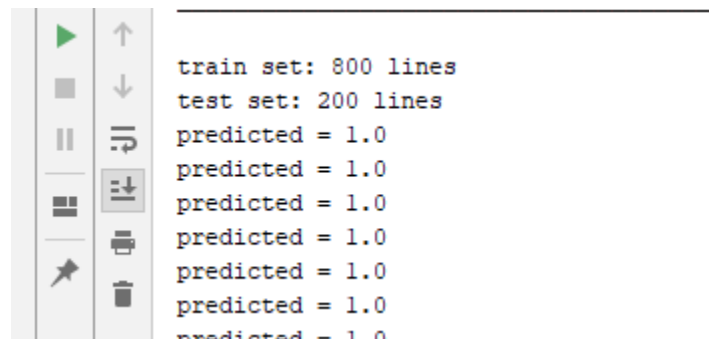
Pada fungsi main ada 2 main() dan main2(). Dimana pada fungsi main() menggunakan DataTrain\_Tugas3\_AI.csv sebagai data train dan data testnya, pada fungsi ini juga terdapat hasil akurasi yaitu sebagai berikut:



```
X1= 1.346162, X2= 0.996949, X3= 0.842938, X4= 1.56318, X5= 3.051993, PREDICTED = '3', ACTUAL= '3'
X1= 1.792138, X2= -1.532045, X3= -0.212688, X4= 1.346853, X5= 3.14493, PREDICTED = '2', ACTUAL= '2'
X1= 1.320487, X2= 0.129181, X3= 3.289529, X4= 0.63392, X5= -0.57673, PREDICTED = '3', ACTUAL= '3'
Accuracy = 88.23529411764706%
```

Gambar 4 Hasil Running Fungsi main() dengan akurasi

Sedangkan pada fungsi main2() merupakan fungsi untuk mencari Y yang belum ada sebelumnya pada DataTest\_Tugas3\_AI.csv dengan data training DataTrain\_Tugas3\_AI.csv. Lalu pada fungsi main2() ini mengekspor hasil prediksi ke dalam TebakanTugas3.csv, hasil dapat dilihat dalam file tersebut.



```
train set: 800 lines
test set: 200 lines
predicted = 1.0
predicted = 1.0
predicted = 1.0
predicted = 1.0
predicted = 1.0
predicted = 1.0
predicted = 1.0
```

Gambar 5 Hasil Running Fungsi main2()