

# Sistem Klasifikasi Data Menggunakan metode K-Nearest Neighbors

Bintang Peryoga

IF-40-04

Fakultas Informatika, Universitas Telkom

Jl. Telekomunikasi no.1 Terusan Buah Batu, Kab. Bandung, Jawa Barat

bintangperyoga.bp@gmail.com

## I. PENDAHULUAN

Laporan ini berisi klasifikasi data menggunakan metode K-Nearest Neighbors dimana 200 data testing dibandingkan dengan 800 data training yang kelas yang telah diketahui

## II. PEMBAHASAN

### A. Analisis Masalah

Terdapat 200 data testing yang belum terklasifikasi dan memiliki 800 data training yang telah terklasifikasi. Tujuan dibuatnya system klasifikasi data ini adalah agar data testing yang masuk dapat terklasifikasi dengan tepat dengan cara membandingkan dengan klasifikasi data yang ada pada data training.

Dengan menggunakan metode K-Nearest Neighbors, kita akan mengambil 5 data selisih teratas yang nantinya akan dipilih data manakah yang hasilnya mendekati kemiripan. Hasil tersebut nantinya disimpan di file dengan nama `TebakanTugas3.csv`.

### B. Strategi Penyelesaian Masalah

- Membuat 2 buah array dari file data testing dan data training

```
# membuat array baru dari data Test
datafile = open("DataTest_Tugas3_AI.csv", "r")
data = csv.reader(datafile)
dataTest = []
for row in data:
    dataTest.append(row)

# membuat array baru dari data Train
datafile = open("DataTrain_Tugas3_AI.csv", "r")
data = csv.reader(datafile)
dataTrain = []
for row in data:
    dataTrain.append(row)
```

Gambar 1.1

Pada awal pengerjaan, kita memiliki data testing dan data training dalam bentuk .csv, dari file tersebut kita convert ke array agar dapat terbaca pada saat program dijalankan.

- Penentuan Nilai K

# nilai K = 5

Gambar 1.2 nilai K

Penentuan nilai K didapatkan dengan metode Validation Split. Dimana kita membagi data training menjadi 2 bagian, sehingga kita mempunyai data training 1 dan 2. Setelah itu kita cari selisihnya dan tentukan nilai K mana yang akurasi paling optimum.

- Pembuatan array

```
hasil = []
selisih = []
temp = [0, 0, 0, 0, 0]
```

Gambar 1.3

Array hasil untuk menampung hasil akhir kelas data testing yang dimasukkan, array selisih untuk menampung total penjumlahan selisih dari tiap atribut data training yang dikurangi dengan data testing, dan array temp untuk menampung 5 data array selisih teratas setelah disorting secara ascending.

## iv. Klasifikasi

```

adayangsama = False
for p in range(0,5):
    temp[p] = selisih[p][1]
for q in range(1,5):
    if selisih[0][0] == selisih[p][0]:
        if selisih[0][1] != selisih[p][1]:
            adayangsama = True
if adayangsama == False:
    hasil.append(selisih[0][1])
elif adayangsama == True:
    hasil.append(most_common(temp))
del selisih[0:800]

```

Gambar 1.4

Lima data teratas diambil, jika data terkecil atau data ke-1 selisihnya tidak ada yang sama, maka data ke-1 diambil dan dimasukkan ke array hasil. Tetapi jika selisihnya ada yang sama, maka menggunakan fungsi `most_common` yang tujuannya mengambil angka terbanyak yang muncul.

## v. Penyimpanan Hasil Klasifikasi

```

# hasil diconvert ke file .csv
datahasil = open('TebakanTugas3.csv', 'w')
datahasil = csv.writer(datahasil, lineterminator='\n')
for d in hasil:
    datahasil.writerow([d])
datahasil.close()

```

Gambar 1.5

Hasil klasifikasi yang telah didapatkan diubah bentuknya dari hasil yang berupa array menjadi `TebakanTugas3` yang berupa `.csv`.

## C. Hasil Running

Dari Program yang telah dibuat, terklasifikasi lah data testing yang sebelumnya belum memiliki kelas. Klasifikasi data tersebut otomatis tersimpan di file `TebakanTugas3.csv` ketika program dijalankan.

## III. KESIMPULAN

Kesimpulan dari laporan ini, metode K-Nearest Neighbors dapat menjadi salah satu solusi untuk sistem klasifikasi data. Penentuan nilai K juga mempengaruhi hasil yang akan diklasifikasikan dengan data testing.

## IV. REFERENSI

- [1] Krassowski, 2018, *Find The Most Common Element in a List*, [online], (<https://stackoverflow.com/questions/1518522/find-the-most-common-element-in-a-list> diakses tanggal 1 Desember 2018)

