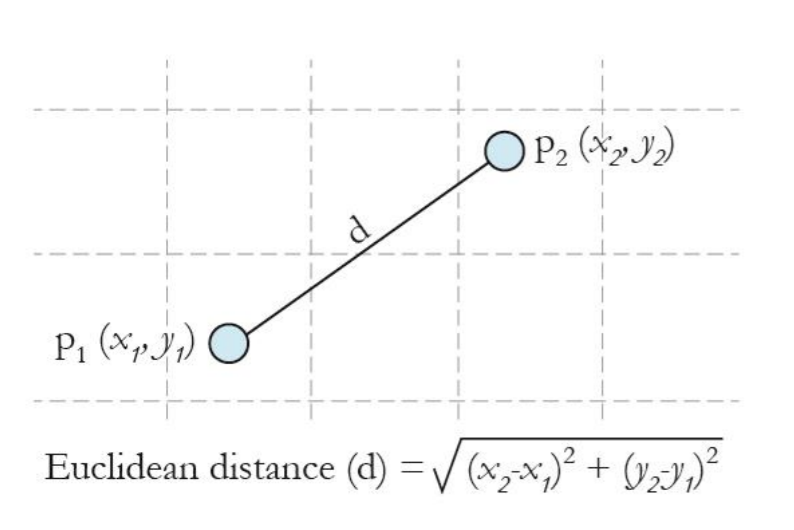
**Praktikum 5**

**K-Nearest Neighbor (KNN)**

**Pendahuluan**

* Dapat digunakan untuk tujuan klasifikasi
* Tidak menyusun model atau mengekstrak aturan logika tertentu sebagai hasil dari analisis
* Identikasi k buah individu tetangga terdekat dilakukan dengan terlebih dahulu menghitung jarak dari individu yang akan diduga dengan setiap individu yang ada pada gugus data training. Jika ini sudah dilakukan maka tinggal mencari k buah amatan yang jaraknya paling kecil. Penghitungan jarak dari dua amatan A dan B dapat menggunakan formula Euclid distance



Dataset 1 :

|  |  |  |
| --- | --- | --- |
| x | y | kelas |
| 0.15 | 0 | 1 |
| 0.3 | 0.1 | 1 |
| 0.11 | 0.12 | 1 |
| 0.04 | 0.13 | 1 |
| 0.18 | 0.17 | 1 |
| 0.1 | 0.19 | 1 |
| 0.54 | 0.2 | 2 |
| 0.19 | 0.21 | 1 |
| 0.6 | 0.24 | 2 |
| 0.66 | 0.25 | 2 |
| 0.33 | 0.26 | 2 |
| 0.17 | 0.27 | 1 |
| 0.52 | 0.29 | 2 |
| 0.52 | 0.33 | 2 |
| 0.49 | 0.35 | 2 |
| 0.68 | 0.36 | 2 |
| 0.55 | 0.43 | 2 |
| 0.37 | 0.44 | 2 |
| 0.52 | 0.47 | 2 |
| 0.58 | 0.5 | 2 |

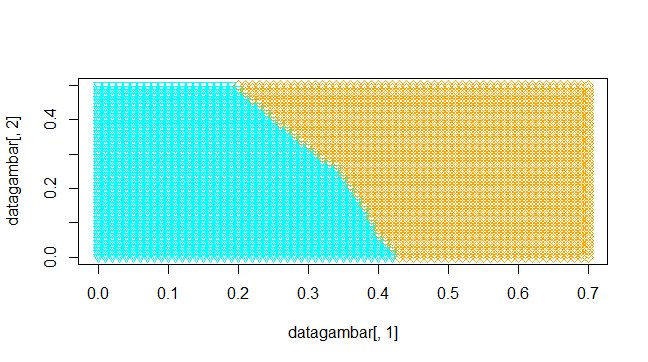
|  |  |  |
| --- | --- | --- |
|  |  |  |
| x | y | kelas |
| 0.1 | 0.2 | ? |
| 0.4 | 0.1 | ? |

|  |
| --- |
| > data.training <- dataku[,1:2]  > View(data.training)  > kelas <- as.factor(dataku[,3])  > View(dataku)  > str(dataku)  'data.frame': 20 obs. of 4 variables:  $ x : num 0.15 0.3 0.11 0.04 0.18 0.1 0.54 0.19 0.6 0.66 ...  $ y : num 0 0.1 0.12 0.13 0.17 0.19 0.2 0.21 0.24 0.25 ...  $ kelas : int 1 1 1 1 1 1 2 1 2 2 ...  $ kelas1: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 2 2 ...  > kelas <- as.factor(dataku[,3])  > amatan.baru <- c(0.1, 0.2)  > library(class)  > knn(data.training, amatan.baru, kelas, k=5)  [1] 1  Levels: 1 2  > amatan.baru2 <- c(0.4, 0.1)  2  > knn(data.training, amatan.baru2, kelas, k=5)  [1] 2  Levels: 1 2 |

Rscript

|  |
| --- |
| x <- seq(0.00, 0.70, by=0.01)  y <- seq(0.00, 0.50, by=0.01)  grid <- NULL  for (i in x) {  for (j in y) {  grid <- rbind(grid, c(i, j))  }  }  datagambar <- data.frame(grid) |

|  |
| --- |
| > qplot(datagambar$X2, datagambar$X1)    > prediksi <- knn(data.training, datagambar, kelas, k = 5)  > plot (datagambar[,1], datagambar[,2],  + col=ifelse(prediksi == "1", "cyan", "orange"),  + pch=ifelse(prediksi == "1", 6, 5)) |



Perhitungan excel dataset 1 :

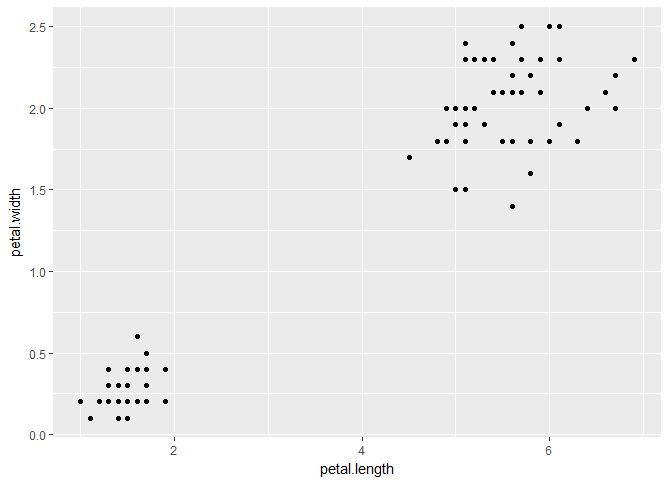
|  |
| --- |
|  |

Dataset 2 : iris.csv

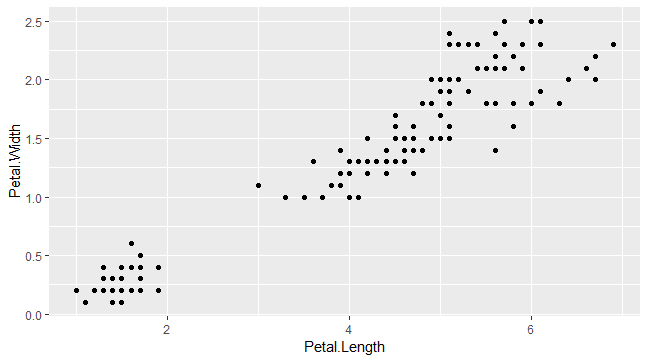
> dataku=read.delim("clipboard")

> library(ggplot2)

> ggplot(dataku, aes(x=petal.length, y=petal.width)) + geom\_point()



> ggplot(iris, aes(x=Petal.Length, y=Petal.Width)) + geom\_point()



|  |
| --- |
| > kelas\_iris <- as.factor(iris[,4])  > pengamatan.baru\_iris <- c(3.8, 1.5)  > library(class)  > data.training\_iris <- iris[,3:4]  > kelas\_iris <- as.factor(iris[,5])  > knn(data.training\_iris, pengamatan.baru\_iris, kelas\_iris, k=5)  [1] versicolor  Levels: setosa versicolor virginica |
|  |

|  |
| --- |
| x <- seq(1.00, 6.00, by=0.2)  y <- seq(0.00, 2.50, by=0.05)  grid <- NULL  for (i in x) {  for (j in y) {  grid <- rbind(grid, c(i, j))  }  }  datagambar <- data.frame(grid) |
| > qplot(datagambar$X1, datagambar$X2)    > prediksi <- knn(data.training, datagambar, kelas, k = 5)  > plot (datagambar[,1], datagambar[,2],col=ifelse(prediksi == "1", "cyan", "orange"), pch=ifelse(prediksi == "1", 6, 5)) |

Perhitungan excel dataset 2 :

|  |
| --- |
|  |