TUGAS BAGIAN II

SK-5222 PENAMBANGAN DATA DALAM SAINS

Mohammad Rizka Fadhli - 20921004

21 Mei 2022

Contents

PENDAHULUAN **5**

DASAR TEORI **5**

K-Means Clustering 5

Bahasa Pemrograman yang Digunakan 6

SOAL DAN PEMBAHASAN **7**

Soal I 7

Pertanyaan 7

Pembahasan 7

K-Means Clustering pada pasangan titik *random* I 13

K-Means Clustering pada pasangan titik *random* II 16

K-Means Clustering pada pasangan titik *random* III 18

K-Means Clustering pada pasangan titik *random* IV 20

K-Means Clustering pada pasangan titik *random* V 22

K-Means Clustering pada pasangan titik *random* VI 24

K-Means Clustering pada pasangan titik *random* VII 26

K-Means Clustering pada pasangan titik *random* VIII 28

K-Means Clustering pada pasangan titik *random* IX 30

K-Means Clustering pada pasangan titik *random* X 32

Kesimpulan dari 10 *Clustering* 34

Soal II 35

Pertanyaan 35

Pembahasan 35

PENUTUP **37**

List of Figures

1 Flowchart Pengerjaan K-Means Clustering 8
2 Scatterplot dari Data 9
3 Hasil Clustering I 15
4 Hasil Clustering II 17
5 Hasil Clustering III 19
6 Hasil Clustering IV 21
7 Hasil Clustering V 23
8 Hasil Clustering VI 25
9 Hasil Clustering VII 27
10 Hasil Clustering VIII 29
11 Hasil Clustering IX 31
12 Hasil Clustering X 33

List of Tables

1 Data Soal I 7
12 Rekap Hasil 10 Kali Clustering 34
13 Data Soal II 35
14 Hasil Perhitungan Entropy dan Purity 36

PENDAHULUAN

Salah satu analisa *unsupervised learning* yang biasa dilakukan adalah *clustering*. Inti analisa ini adalah mengelompokkan sekumpilan data berdasarkan *similarity* yang ada pada masing-masing atributnya.

Ada berbagai metode yang bisa digunakan, yakni:

1. *K-means clustering*,
2. *Hierarchical clustering*,
3. *DBScan clustering*,
4. *Fuzzy c-means*,
5. dan lainnya.

Masing-masing metode *clustering* tersebut memiliki keunggulan dan kelemahan tersendiri. Oleh karena itu, pemilihan metode yang tepat akan menentukan keberhasilan dan ketepatan *clustering*.

Validitas dari *clustering* bisa diukur menggunakan berbagai macam cara. Beberapa di antaranya adalah dengan menghitung:

1. *Squared standard error* (SSE),
2. *Silhouette coeeficient*,
3. *Purity* dan *entropy* dari masing-masing *cluster*.

DASAR TEORI

K-Means Clustering

K-means clustering merupakan pengelompokkan berdasarkan partisi di mana *input* yang harus diketahui adalah banyaknya *clusters*. Dari *input* tersebut, akan dilakukan iterasi pencarian sentroid (pusat *cluster*) hingga konvergen. Algoritmanya adalah sebagai berikut:

```
Langkah I
  Pilih K titik sembarang sebagai sentroid awal
Langkah II
  Hitung jarak semua titik data ke sentroid
  Assign titik tersebut ke sentroid terdekatnya
Langkah III
  Update sentroid ke titik terbaru
Langkah IV
  Ulangi langkah II dan III hingga konvergen
```

Bahasa Pemrograman yang Digunakan

Untuk mengerjakan tugas ini, saya menggunakan bahasa **R** dengan algoritma yang dibuat sendiri dengan prinsip *tidy* menggunakan operator `%>%`.

Libraries yang digunakan adalah:

1. `dplyr` untuk *data carpentry*.
2. `ggplot2` untuk visualisasi data.

Tugas ini ditulis menggunakan **R *markdown*** sehingga semua kode pemrograman bisa dilihat langsung di *chunks* masing-masing.

SOAL DAN PEMBAHASAN

Soal I

Diberikan 10 buah titik data sebagai berikut:

Table 1: Data Soal I

titik	x	y
p1	4.0	5.2
p2	2.1	3.9
p3	3.4	3.1
p4	2.7	2.0
p5	0.8	4.1
p6	4.6	2.9
p7	4.3	1.2
p8	2.2	1.0
p9	4.1	4.1
p10	1.5	3.0

- Lakukan klasterisasi dari data tersebut dengan menggunakan algoritma *k-means* dengan jumlah partisi $K = 2$ sebanyak 10 kali.
- Tentukan sentroid awal (secara *random*) yang berbeda setiap melakukan klasterisasi.
- *Stopping criteria* untuk klasterisasi bisa ditentukan sendiri (tidak harus sampai tidak ada perubahan sentroid)

Pertanyaan

1. Tuliskan hasil akhir kluster yang didapat untuk setiap klasterisasi!
2. Hitung nilai *average* ***SSE*** untuk masing-masing hasil klusterisasi!
3. Hitung nilai *average* ***Sillhouette Coefficient*** untuk masing-masing hasil klusterisasi!
4. Dari hasil ***SSE*** dan ***Sillhouette Coefficient***, menurut Anda, hasil klasterisasi mana yang memberikan hasil terbaik? Berikan alasannya!
5. Apakah algoritma *K-means* sudah memberikan hasil yang baik? Apa yang dapat dilakukan agar hasil klasterisasi lebih baik?

Pembahasan

Untuk melakukan *k-means clustering* ini, saya akan membuat algoritma sendiri dengan menggunakan 2 titik *random* dan akan dilakukan sebanyak 10 kali. Berikut ini adalah *flowchart* dari algoritma tersebut:

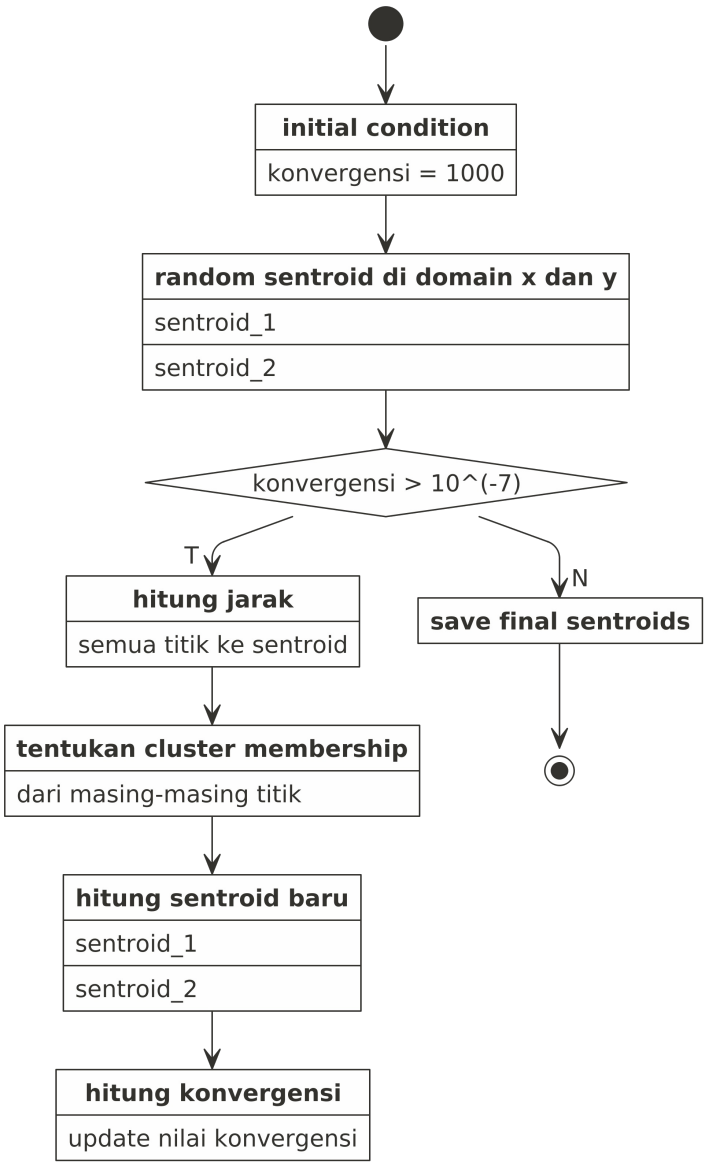

Figure 1: Flowchart Pengerjaan K-Means Clustering

Sebagai pengingat, algoritma *k-means clustering* dilakukan secara iteratif dengan mengandalkan suatu *stopping criteria* tertentu. Pada tugas ini, *stopping criteria* yang saya gunakan adalah sebagai berikut:

$$\text{konvergensi} = \sqrt{(x_1^{(k+1)} - x_1^{(k)})^2 + (x_2^{(k+1)} - x_2^{(k)})^2}$$

Dimana $x_1^{(k)}$ dan $x_2^{(k)}$ menandakan sentroid 1 dan 2 pada iterasi ke - k.

Sebagai gambaran, berikut adalah *scatterplot* dari data soal tersebut:

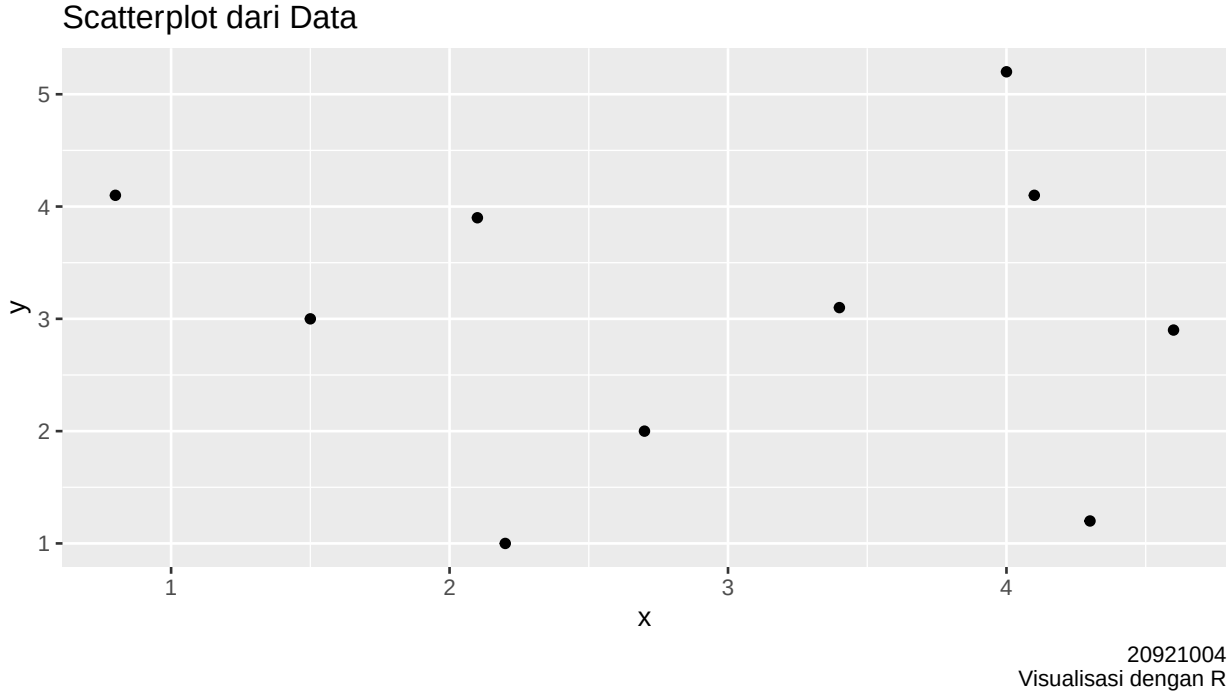

Figure 2: Scatterplot dari Data

Untuk menyelesaikan permasalahan ini, saya akan membuat beberapa program sebagai berikut:

Program untuk membuat sepasang titik secara *random*

```
# program untuk membuat titik sentroid secara random
random_titik = function(){
  list(
    sentroid_1 = runif(2,0,6),
    sentroid_2 = runif(2,0,6)
  )
}
```

Program menghitung *euclidean distance*

```
# program untuk menghitung jarak
jarak = function(x1,x2){
  sb_1 = (x1[1] - x2[1])^2
  sb_2 = (x1[2] - x2[2])^2
  sqrt(sb_1 + sb_2)
}
```

Program untuk menghitung sentroid baru hasil iterasi ke-i

```
# program untuk menghitung sentroid baru
new_sentroid = function(data){
  hit =
    data %>%
    group_by(cluster_no) %>%
    summarise(x = mean(x),
              y = mean(y)) %>%
    ungroup()
  output = list(sentroid_1 = c(hit$x[1],hit$y[1]),
                sentroid_2 = c(hit$x[2],hit$y[2]))
  return(output)
}
```

Program untuk menghitung konvergensi

```
# program untuk menghitung selisih sentroid baru dengan sentroid lama
konvergen_yn = function(){
  part1 = sentroid_baru$sentroid_1 - sentroid_1
  part2 = sentroid_baru$sentroid_2 - sentroid_2
  sqrt(sum(part1^2) + sum(part2^2))
}
```

Program untuk menghitung SSE

```
hitung_SSE = function(df){
  # hitung jarak terhadap sentroid
  for(i in 1:nrow(df)){
    titik = c(df$x[i],df$y[i])
    df$jarak_sentroid1[i] = jarak(titik,sentroid_1)
    df$jarak_sentroid2[i] = jarak(titik,sentroid_2)
  }
  SSE_final =
    df %>%
    mutate(jarak_thd_centroid = ifelse(cluster_no == 1,
                                       jarak_sentroid1,
                                       jarak_sentroid2))
  SSE_final$jarak_thd_centroid^2 %>% sum() %>% round(4)
}
```

Program untuk menghitung *Silhouette Coefficient*

```
sil_coeff = function(df){
  # menghitung distance matrix
  tes =
    df %>%
    select(x,y)
  mat_dist = dist(tes,upper = T) %>% as.matrix()
  # mengambil id titik per cluster
  id_cl_1 = which(df_1$cluster_no == 1)
  id_cl_2 = which(df_1$cluster_no == 2)
  # menghitung nilai a
  a1 = mat_dist[id_cl_1,id_cl_1] %>% mean()
  a2 = mat_dist[id_cl_2,id_cl_2] %>% mean()
```

```
  a = mean(a1,a2)
  # menghitung nilai b
  b = rep(NA,10)
  for(i in 1:10){
    if(i %in% id_cl_1){
      b[i] = mat_dist[i,id_cl_2] %>% mean()
    } else
    if(i %in% id_cl_2){
      b[i] = mat_dist[i,id_cl_1] %>% mean()
    }
  }
  b = min(b)
  # menghitung silhouette coefficient
  s_coeff = (b-a)/max(a,b)
  return(s_coeff)
}
```

K-Means Clustering pada pasangan titik *random* I

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 2.303229 5.946158
##
## $sentroid_2
## [1] 5.666725 1.667516
```

Saya akan lakukan *clustering* berikut ini:

```
# initial sentroid
sentroid_1 = random$sentroid_1
sentroid_2 = random$sentroid_2
# menyiapkan template untuk menghitung jarak
df$jarak_sentroid1 = NA
df$jarak_sentroid2 = NA
# initial konvergensi
konvergensi = 1000
# proses iterasi k-means hingga konvergensi tercapai
while(konvergensi > 10^(-7)){
  # hitung jarak terhadap sentroid
  for(i in 1:nrow(df)){
    titik = c(df$x[i],df$y[i])
    df$jarak_sentroid1[i] = jarak(titik,sentroid_1)
    df$jarak_sentroid2[i] = jarak(titik,sentroid_2)
  }
  # memasukkan masing-masing titik ke cluster terdekat
  df =
    df %>%
    mutate(cluster_no = ifelse(jarak_sentroid1 < jarak_sentroid2,1,2))
  # menghitung sentroid baru
  sentroid_baru = new_sentroid(df)
  # menghitung konvergensi
  konvergensi = konvergen_yn()
  # update sentroid baru
  sentroid_1 = sentroid_baru$sentroid_1
  sentroid_2 = sentroid_baru$sentroid_2
}
```

```r
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.8840382	3.2092367	1
p2	2.1	3.9	0.4308132	2.2924223	1
p3	3.4	3.1	1.3159027	1.0607544	2
p4	2.7	2.0	2.0696860	0.7410803	2
p5	0.8	4.1	1.7004705	3.3486117	1
p6	4.6	2.9	2.3990832	1.4440222	2
p7	4.3	1.2	3.3792899	1.2021647	2
p8	2.2	1.0	3.0746707	1.6183943	2
p9	4.1	4.1	1.6004999	2.1631459	1
p10	1.5	3.0	1.4572577	2.1645323	1

```r
# sentroid 1
sentroid_1
```

```
## [1] 2.50 4.06
```

```r
# sentroid 2
sentroid_2
```

```
## [1] 3.44 2.04
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 19.136
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.08707463
```

Berikut adalah grafik hasilnya:

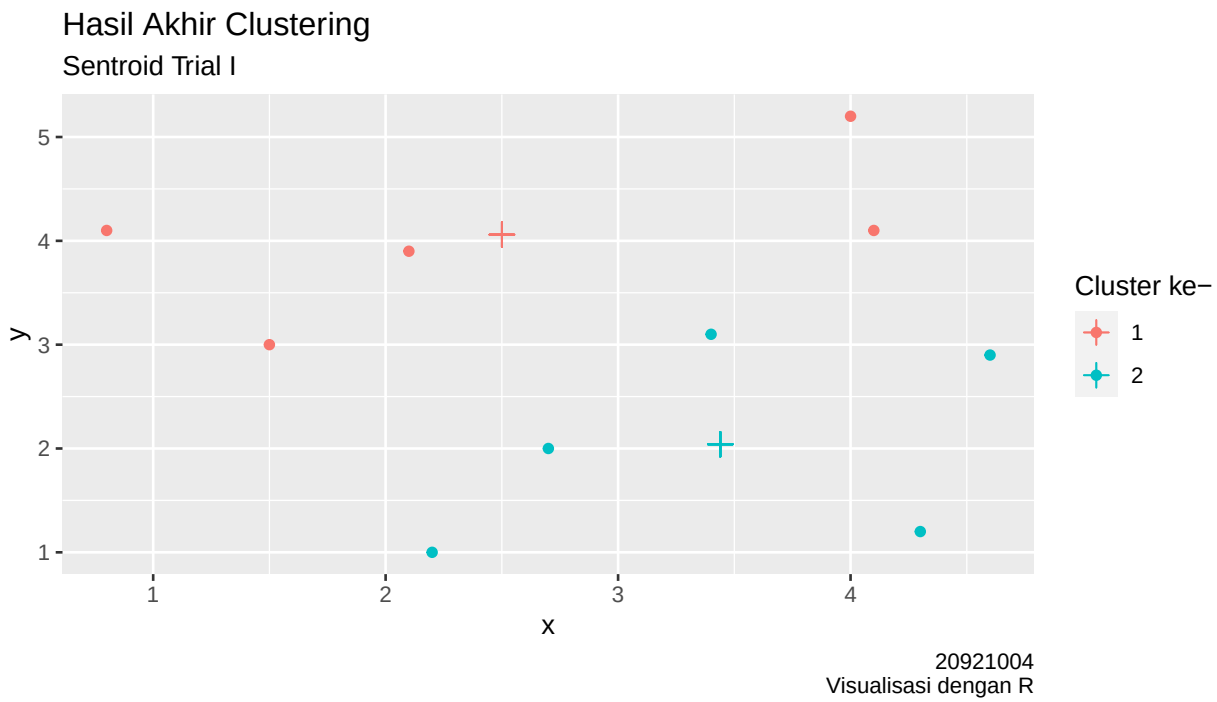

Figure 3: Hasil Clustering I

Dengan prinsip yang sama, saya akan ulangi proses di atas hingga 10 kali.

K-Means Clustering pada pasangan titik *random* II

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 3.622327 3.829856
##
## $sentroid_2
## [1] 2.433041 3.375570
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.9016835	3.2155248	1
p2	2.1	3.9	2.0689128	1.1258774	2
p3	3.4	3.1	0.7088018	1.5689487	1
p4	2.7	2.0	1.8958903	1.1600000	2
p5	0.8	4.1	3.3761517	1.6773789	2
p6	4.6	2.9	0.6560488	2.7418242	1
p7	4.3	1.2	2.1114924	2.9178074	1
p8	2.2	1.0	2.9705892	1.8318297	2
p9	4.1	4.1	0.8002500	2.5899035	1
p10	1.5	3.0	2.5973833	0.4118252	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.08 3.30
```

```
# sentroid 2
sentroid_2
```

```
## [1] 1.86 2.80
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 18.6
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.1889305
```

Berikut adalah grafik hasilnya:

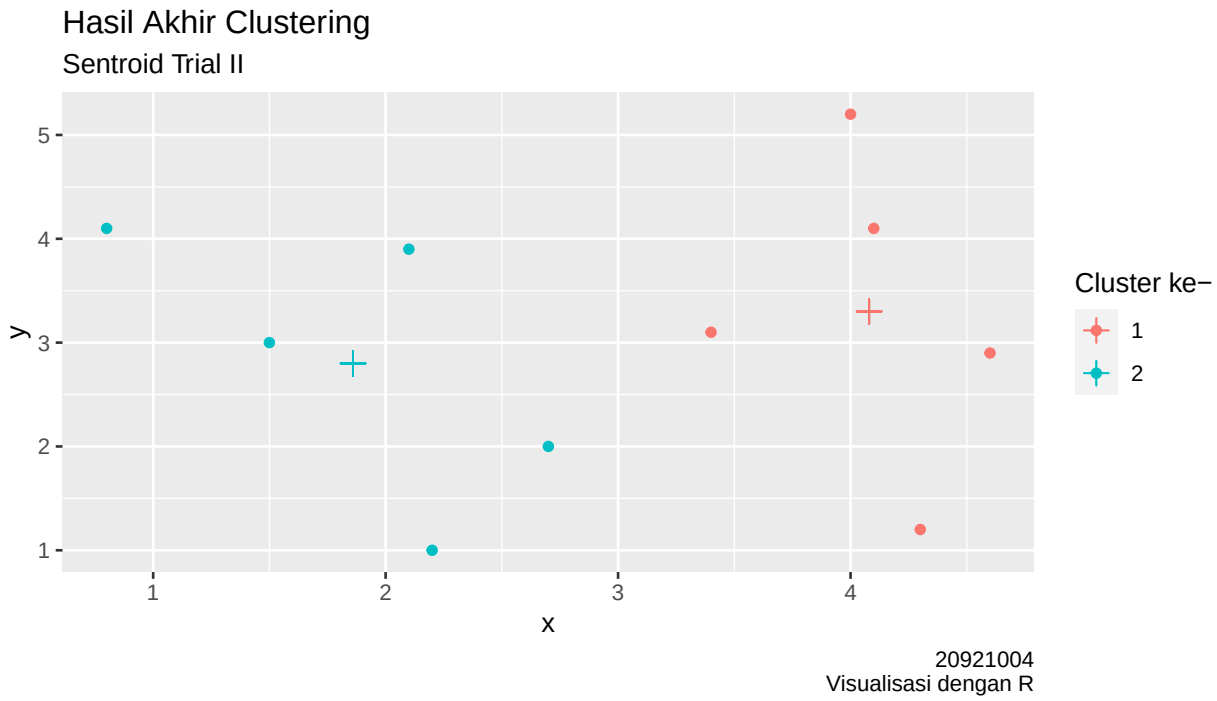

Figure 4: Hasil Clustering II

K-Means Clustering pada pasangan titik *random* III

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 5.533929 5.744867
##
## $sentroid_2
## [1] 2.048983 2.196384
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.1571037	3.0257736	1
p2	2.1	3.9	2.1398338	1.3270344	2
p3	3.4	3.1	1.2762793	1.0860902	2
p4	2.7	2.0	2.5733679	0.6715805	2
p5	0.8	4.1	3.4334951	2.2044482	2
p6	4.6	2.9	1.2229291	2.1901449	1
p7	4.3	1.2	2.8674418	2.3457299	2
p8	2.2	1.0	3.6795229	1.6303875	2
p9	4.1	4.1	0.1374369	2.2362961	1
p10	1.5	3.0	2.9340908	1.0054951	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.233333 4.066667
```

```
# sentroid 2
sentroid_2
```

```
## [1] 2.428571 2.614286
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 20.2762
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.3102129
```

Berikut adalah grafik hasilnya:

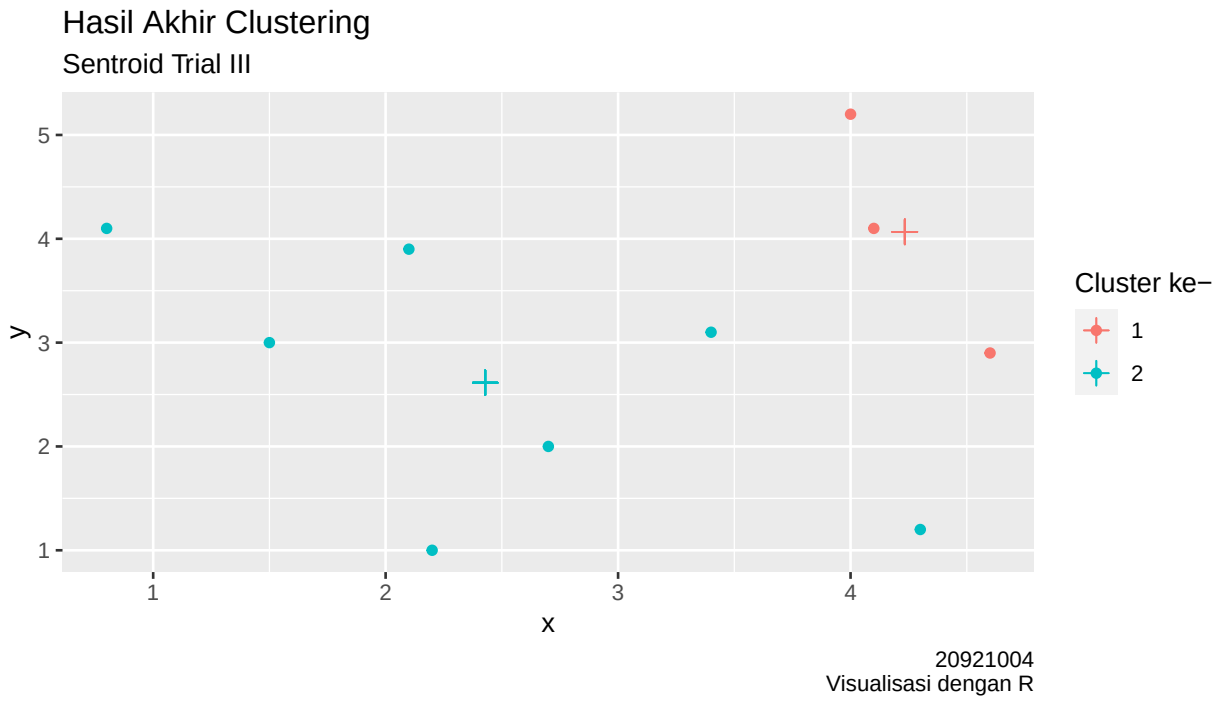

Figure 5: Hasil Clustering III

K-Means Clustering pada pasangan titik *random* IV

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 5.496974 5.192821
##
## $sentroid_2
## [1] 0.9435708 0.5504905
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.3752273	3.1804961	1
p2	2.1	3.9	1.9264605	1.3767918	2
p3	3.4	3.1	0.9572095	1.2671052	1
p4	2.7	2.0	2.2552716	0.6871843	2
p5	0.8	4.1	3.2367036	2.1460558	2
p6	4.6	2.9	1.0891510	2.3619672	1
p7	4.3	1.2	2.6393655	2.4315062	2
p8	2.2	1.0	3.3632202	1.5347819	2
p9	4.1	4.1	0.2850439	2.4115463	1
p10	1.5	3.0	2.6563603	0.8975275	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.025 3.825
```

```
# sentroid 2
sentroid_2
```

```
## [1] 2.266667 2.533333
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 20.1217
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.4142629
```

Berikut adalah grafik hasilnya:

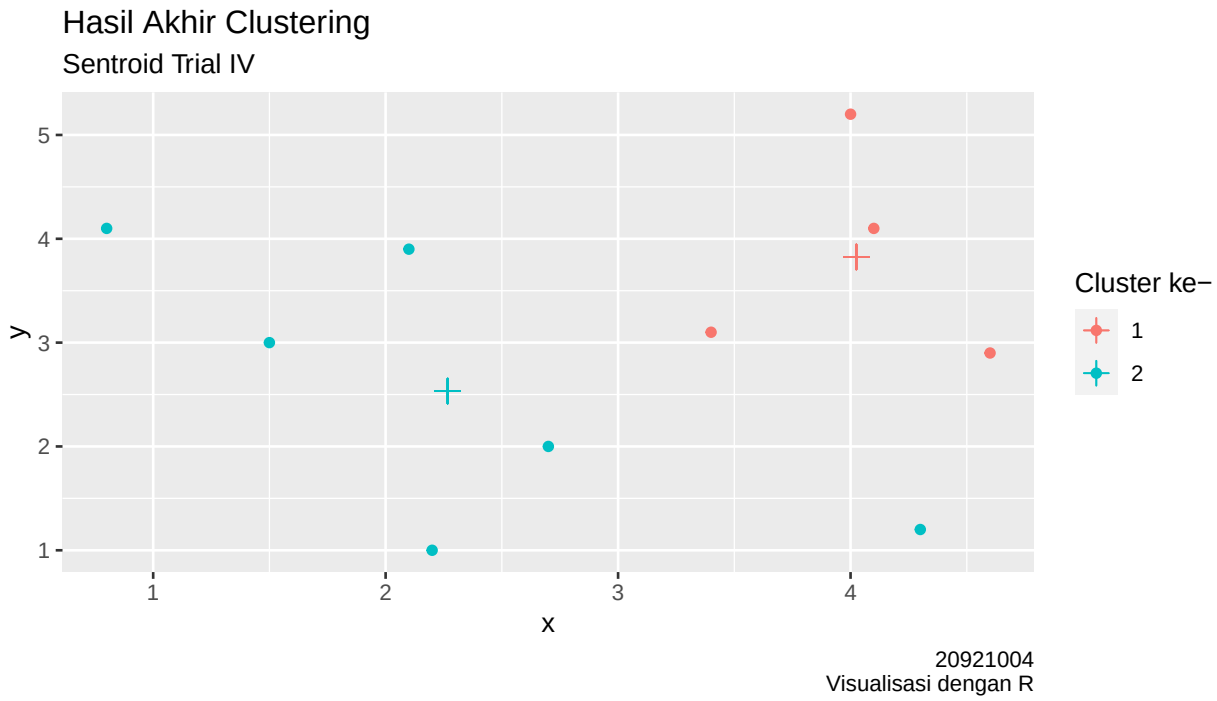

Figure 6: Hasil Clustering IV

K-Means Clustering pada pasangan titik *random* V

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 3.109427 3.607948
##
## $sentroid_2
## [1] 3.6598941 0.7484456
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.7971633	3.9129415	1
p2	2.1	3.9	0.8407965	2.6803814	1
p3	3.4	3.1	0.8087531	1.7323715	1
p4	2.7	2.0	1.7719469	0.7031674	2
p5	0.8	4.1	2.1560073	3.5253053	1
p6	4.6	2.9	1.8783949	2.1450201	1
p7	4.3	1.2	2.9016885	1.2494443	2
p8	2.2	1.0	2.8517807	0.9545214	2
p9	4.1	4.1	1.2205720	2.8909821	1
p10	1.5	3.0	1.6168122	2.2392955	1

```
# sentroid 1
sentroid_1
```

```
## [1] 2.928571 3.757143
```

```
# sentroid 2
sentroid_2
```

```
## [1] 3.066667 1.400000
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 19.8381
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.02781838
```

Berikut adalah grafik hasilnya:

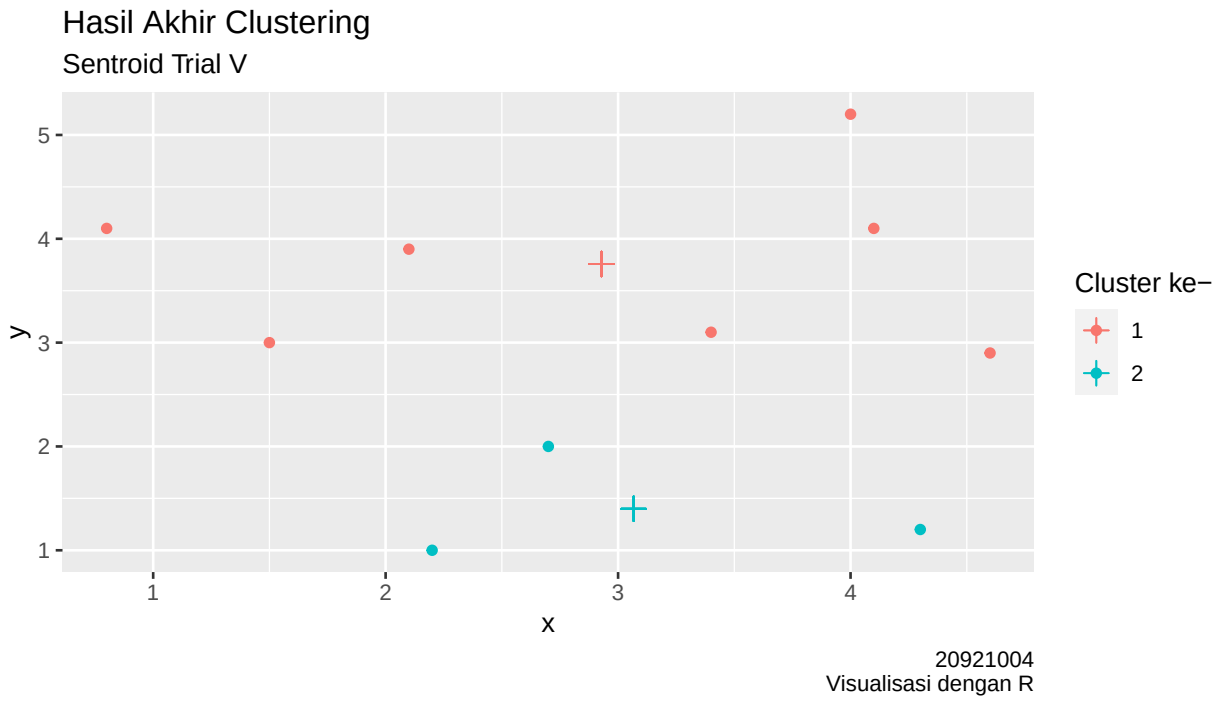

Figure 7: Hasil Clustering V

K-Means Clustering pada pasangan titik *random* VI

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 4.378696 2.017035
##
## $sentroid_2
## [1] 0.6586512 2.2695625
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.9016835	3.2155248	1
p2	2.1	3.9	2.0689128	1.1258774	2
p3	3.4	3.1	0.7088018	1.5689487	1
p4	2.7	2.0	1.8958903	1.1600000	2
p5	0.8	4.1	3.3761517	1.6773789	2
p6	4.6	2.9	0.6560488	2.7418242	1
p7	4.3	1.2	2.1114924	2.9178074	1
p8	2.2	1.0	2.9705892	1.8318297	2
p9	4.1	4.1	0.8002500	2.5899035	1
p10	1.5	3.0	2.5973833	0.4118252	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.08 3.30
```

```
# sentroid 2
sentroid_2
```

```
## [1] 1.86 2.80
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 18.6
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.1889305
```

Berikut adalah grafik hasilnya:

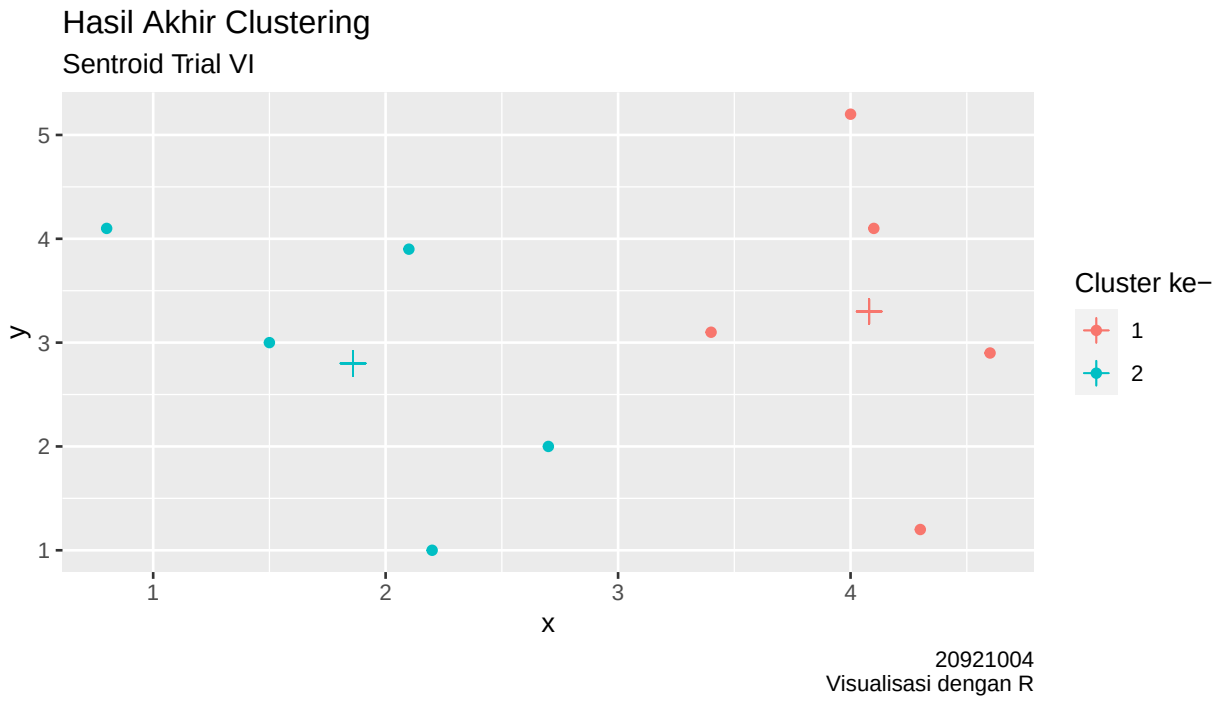

Figure 8: Hasil Clustering VI

K-Means Clustering pada pasangan titik *random* VII

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 5.310390 4.924674
##
## $sentroid_2
## [1] 1.014857 1.814161
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.3752273	3.1804961	1
p2	2.1	3.9	1.9264605	1.3767918	2
p3	3.4	3.1	0.9572095	1.2671052	1
p4	2.7	2.0	2.2552716	0.6871843	2
p5	0.8	4.1	3.2367036	2.1460558	2
p6	4.6	2.9	1.0891510	2.3619672	1
p7	4.3	1.2	2.6393655	2.4315062	2
p8	2.2	1.0	3.3632202	1.5347819	2
p9	4.1	4.1	0.2850439	2.4115463	1
p10	1.5	3.0	2.6563603	0.8975275	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.025 3.825
```

```
# sentroid 2
sentroid_2
```

```
## [1] 2.266667 2.533333
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 20.1217
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.4142629
```

Berikut adalah grafik hasilnya:

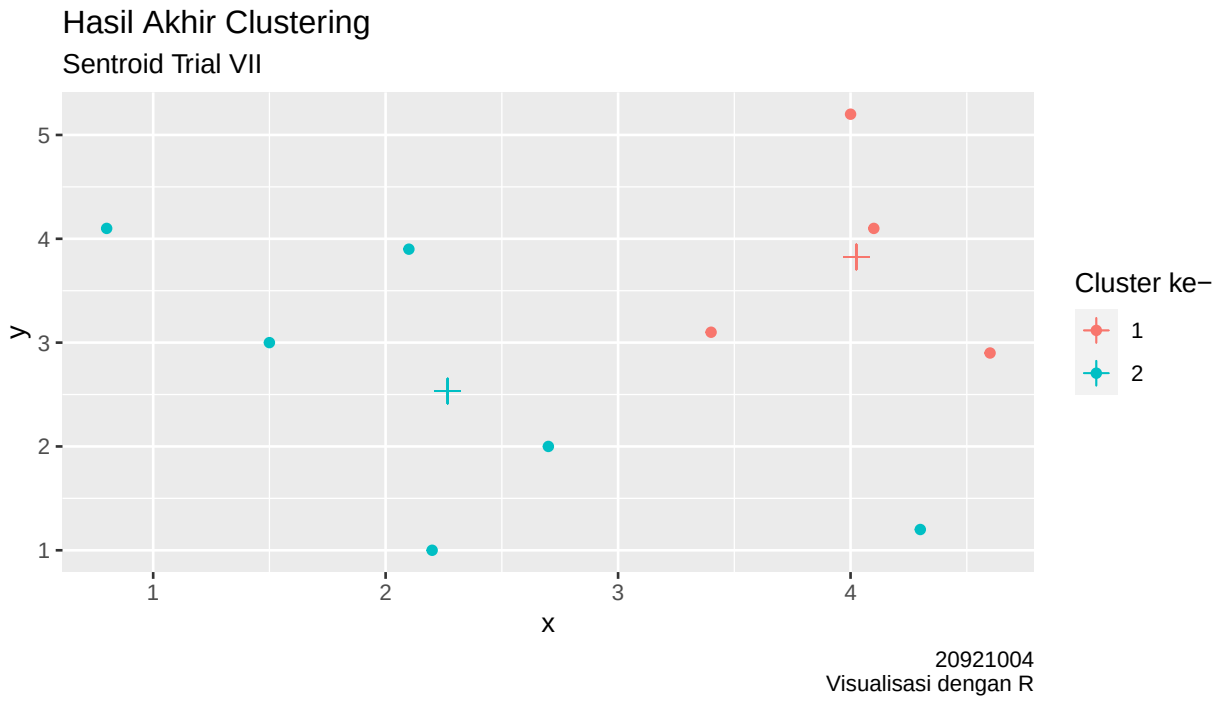

Figure 9: Hasil Clustering VII

K-Means Clustering pada pasangan titik *random* VIII

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 0.2314867 1.0421548
##
## $sentroid_2
## [1] 2.625017 1.307320
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	2.9612310	2.4449031	2
p2	2.1	3.9	0.6749486	1.8800782	1
p3	3.4	3.1	2.0146684	0.3803865	2
p4	2.7	2.0	2.0733762	1.2055145	2
p5	0.8	4.1	0.7951240	3.1060507	1
p6	4.6	2.9	3.2257643	0.9923174	2
p7	4.3	1.2	3.7566238	1.7276267	2
p8	2.2	1.0	2.7656625	2.2779332	2
p9	4.1	4.1	2.6687492	1.4011657	2
p10	1.5	3.0	0.6674995	2.1251170	1

```
# sentroid 1
sentroid_1
```

```
## [1] 1.466667 3.666667
```

```
# sentroid 2
sentroid_2
```

```
## [1] 3.614286 2.785714
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 20.2305
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.6030084
```

Berikut adalah grafik hasilnya:

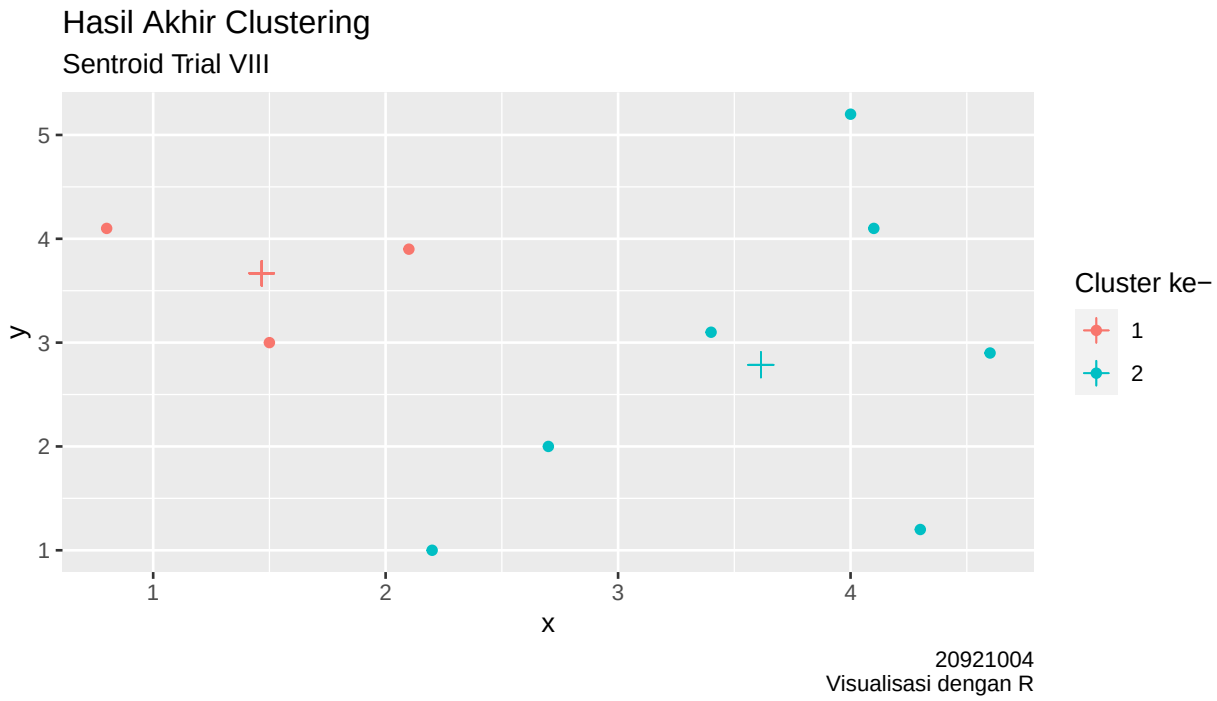

Figure 10: Hasil Clustering VIII

K-Means Clustering pada pasangan titik *random* IX

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 3.583279 2.897936
##
## $sentroid_2
## [1] 0.3276318 1.0728837
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.7971633	3.9129415	1
p2	2.1	3.9	0.8407965	2.6803814	1
p3	3.4	3.1	0.8087531	1.7323715	1
p4	2.7	2.0	1.7719469	0.7031674	2
p5	0.8	4.1	2.1560073	3.5253053	1
p6	4.6	2.9	1.8783949	2.1450201	1
p7	4.3	1.2	2.9016885	1.2494443	2
p8	2.2	1.0	2.8517807	0.9545214	2
p9	4.1	4.1	1.2205720	2.8909821	1
p10	1.5	3.0	1.6168122	2.2392955	1

```
# sentroid 1
sentroid_1
```

```
## [1] 2.928571 3.757143
```

```
# sentroid 2
sentroid_2
```

```
## [1] 3.066667 1.400000
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 19.8381
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.02781838
```

Berikut adalah grafik hasilnya:

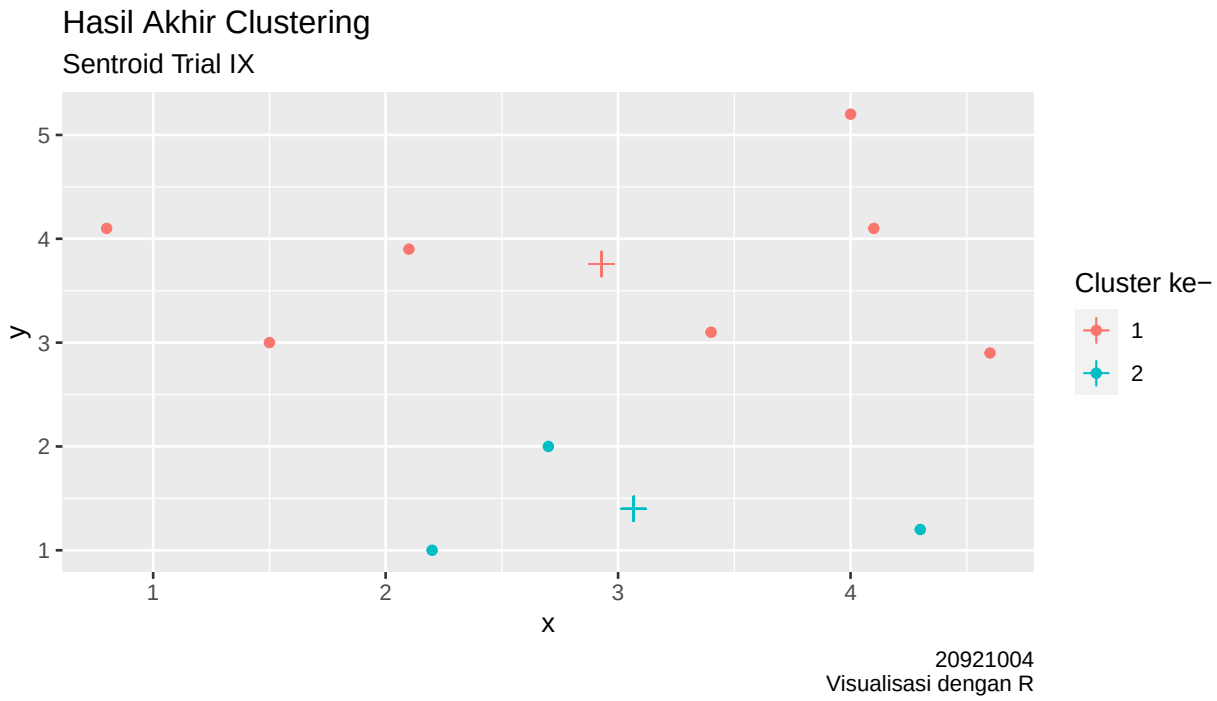

Figure 11: Hasil Clustering IX

K-Means Clustering pada pasangan titik *random* X

Menggunakan titik random berikut ini:

```
random = random_titik()
random
```

```
## $sentroid_1
## [1] 5.607272 2.425094
##
## $sentroid_2
## [1] 1.917969 2.035520
```

Saya akan lakukan *clustering* menghasilkan:

```
# hasil final
df_1 %>% knitr::kable()
```

titik	x	y	jarak_sentroid1	jarak_sentroid2	cluster_no
p1	4.0	5.2	1.9016835	3.2155248	1
p2	2.1	3.9	2.0689128	1.1258774	2
p3	3.4	3.1	0.7088018	1.5689487	1
p4	2.7	2.0	1.8958903	1.1600000	2
p5	0.8	4.1	3.3761517	1.6773789	2
p6	4.6	2.9	0.6560488	2.7418242	1
p7	4.3	1.2	2.1114924	2.9178074	1
p8	2.2	1.0	2.9705892	1.8318297	2
p9	4.1	4.1	0.8002500	2.5899035	1
p10	1.5	3.0	2.5973833	0.4118252	2

```
# sentroid 1
sentroid_1
```

```
## [1] 4.08 3.30
```

```
# sentroid 2
sentroid_2
```

```
## [1] 1.86 2.80
```

Berikut adalah ***SSE*** dari perhitungan ini:

```r
# menghitung SSE
SSE = hitung_SSE(df_1)
SSE
```

```
## [1] 18.6
```

Berikut adalah ***silhouette coefficient*** dari perhitungan ini:

```r
SC = sil_coeff(df_1)
SC
```

```
## [1] 0.1889305
```

Berikut adalah grafik hasilnya:

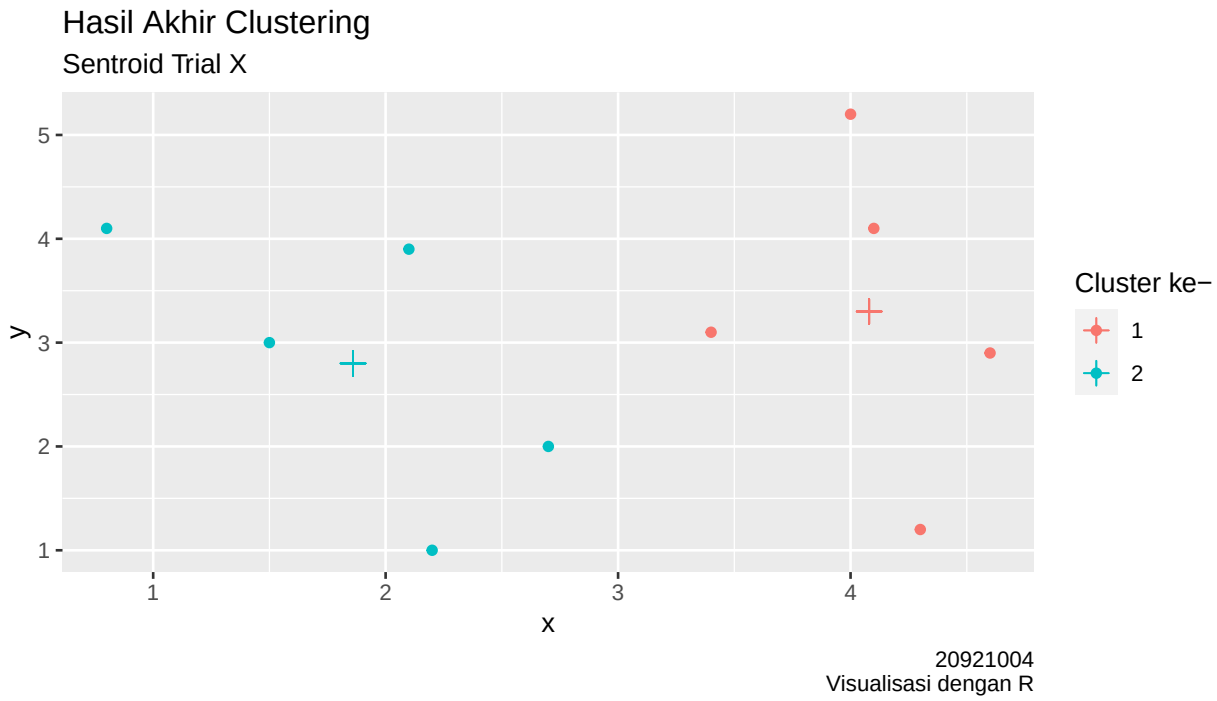

Figure 12: Hasil Clustering X

Kesimpulan dari 10 *Clustering*

Table 12: Rekap Hasil 10 Kali Clustering

cluster	SSE	Sil_Coeff
Cluster 1	19.1360	0.0870746
Cluster 2	18.6000	0.1889305
Cluster 3	20.2762	0.3102129
Cluster 4	20.1217	0.4142629
Cluster 5	19.8381	0.0278184
Cluster 6	18.6000	0.1889305
Cluster 7	20.1217	0.4142629
Cluster 8	20.2305	0.6030084
Cluster 9	19.8381	0.0278184
Cluster 10	18.6000	0.1889305

Jika kita lihat nilai *silhouette coefficient* yang ada, dari 10 kali proses *clustering*, nilai koefisien yang dihasilkan masih rendah (mendekati nol). Selain itu, nilai *SSE* yang ada juga masih relatif besar. Sehingga kita tidak bisa menyimpulkan bahwa *cluster* yang dihasilkan sudah baik.

Soal II

Diberikan *confusion matrix* sebagai berikut:

Table 13: Data Soal II

cluster	entertainment	financial	foreign	metro	national	sports	Total
#1	1	1	0	11	4	676	693
#2	27	89	333	827	253	33	1562
#3	326	465	8	105	16	29	949
Total	354	555	341	943	273	738	3204

Pertanyaan

Hitung nilai *entropy* dan *purity* untuk matriks tersebut! Berikan analisis untuk hasil yang didapat!

Pembahasan

Entropi untuk masing-masing cluster dihitung sebagai berikut:

$$\begin{aligned}\text{Entropy 1} = & -\tfrac{1}{693}\log_2(\tfrac{1}{693}) - \tfrac{1}{693}\log_2(\tfrac{1}{693}) \\ & -0 - \tfrac{11}{693}\log_2(\tfrac{11}{693}) \\ & -\tfrac{4}{693}\log_2(\tfrac{4}{693}) - \tfrac{676}{693}\log_2(\tfrac{676}{693}) \\ & = 0.200\end{aligned}$$

$$\begin{aligned}\text{Entropy 2} = & -\tfrac{27}{1562}\log_2(\tfrac{27}{1562}) - \tfrac{89}{1562}\log_2(\tfrac{89}{1562}) \\ & -\tfrac{333}{1562}\log_2(\tfrac{333}{1562}) - \tfrac{872}{1562}\log_2(\tfrac{872}{1562}) \\ & -\tfrac{253}{1562}\log_2(\tfrac{253}{1562}) - \tfrac{33}{1562}\log_2(\tfrac{33}{1562}) \\ & = 1.841\end{aligned}$$

$$\begin{aligned}\text{Entropy 3} = & -\tfrac{326}{949}\log_2(\tfrac{326}{949}) - \tfrac{465}{949}\log_2(\tfrac{465}{949}) \\ & -\tfrac{8}{949}\log_2(\tfrac{8}{949}) - \tfrac{105}{949}\log_2(\tfrac{105}{949}) \\ & -\tfrac{16}{949}\log_2(\tfrac{16}{949}) - \tfrac{29}{949}\log_2(\tfrac{29}{949}) \\ & = 1.696\end{aligned}$$

Sedangkan untuk *purity* dihitung dengan cara:

$$\begin{aligned}\text{Purity 1} &= \tfrac{676}{693} = 0.975 \\ \text{Purity 2} &= \tfrac{827}{1562} = 0.529 \\ \text{Purity 3} &= \tfrac{465}{949} = 0.490\end{aligned}$$

Total entropy dihitung sebagai berikut:

$$\text{Total entropy} = \frac{693 \times 0.200 + 1562 \times 1.841 + 949 \times 0.490}{3204} = 0.614$$

Total purity dihitung sebagai berikut:

$$\text{Total purity} = \frac{693 \times 0.975 + 1562 \times 0.529 + 949 \times 1.696}{3204} = 1.443$$

Berikut jika disajikan dalam bentuk tabel:

Table 14: Hasil Perhitungan Entropy dan Purity

cluster	entertainment	financial	foreign	metro	national	sports	Total	Entropy	Purity
#1	1	1	0	11	4	676	693	0.200	0.975
#2	27	89	333	827	253	33	1562	1.841	0.529
#3	326	465	8	105	16	29	949	1.696	0.490
Total	354	555	341	943	273	738	3204	0.614	1.443

Dari tabel di atas, kita bisa dapatkan informasi sebagai berikut:

> *Cluster* `#1` memiliki *purity* yang sangat tinggi dan *entropy* terendah. Artinya, cluster ini berhasil mengelompokkan data yang *unique* karakteristiknya (berasal dari satu atribut dominan). Berbeda dengan *cluster* `#2` dan `#3` yang tidak memiliki satu atribut yang dominan. Tapi secara keseluruhan, *cluster* yang dihasilkan sudah bisa memisahkan data menjadi 3 kelompok dengan karakteristik yang berbeda-beda.

PENUTUP

Validitas hasil *clustering* bisa dilihat menggunakan berbagai macam cara, yakni:

1. *SSE*,
2. *Silhouette coefficient*,
3. *Purity*,
4. *Entropy*.