Clustering

Dian Ramadhani

08/01/2020

Clustering

Clustering merupakan teknik untuk menemukan sub-kelompok dalam suatu set data yang memiliki kesamaan karakteristik.

Install Packages

```
# Menginstall package(s)
install.packages("readr") # membaca file
install.packages("ggplot2") # visualisasi data
install.packages("here") # menampilkan direktori
install.packages("cluster") # algoritma clustering
install.packages("factoextra") # visualisasi clustering
install.packages("ggdendro") # visualisasi dendogram
```

Import Library

```
# Mengaktifkan package(s)
library(readr)
library(ggplot2)
library(here)
library(tidyverse)
library(cluster)
library(factoextra)
library(ggdendro)
```

Menampilkan Direktori

```
# Mengetahui direktori proyek
here()
```

Import Data

```
# Mengimport data
df.income <- read_csv(here("data", "raw","clustering_income.csv"))
```

```
## Parsed with column specification:
## cols(
##   income = col_double(),
##   spend = col_double()
## )
```

Data yang digunakan yaitu data “income”. Data ini terdiri atas dua fitur yaitu income dan spend. Data ini digunakan untuk mengelompokkan konsumen berdasarkan karakteristik pendapatan dan pengeluarannya.

Eksplorasi Data

Data yang telah diimpor selanjutnya dieksplorasi untuk mengetahui strukturnya.

```
# Melihat attribute dan struktur data
names(df.income) # menampilkan nama kolom
```

```
## [1] "income" "spend"
```

```
dim(df.income) # menampilkan dimensi tabel
```

```
## [1] 250   2
```

```
head(df.income) # menampilkan beberap data teratas
```

```
## # A tibble: 6 x 2
##   income spend
##    <dbl> <dbl>
## 1    233   150
## 2    250   187
## 3    204   172
## 4    236   178
## 5    354   163
## 6    192   148
```

```
str(df.income) # menampilkan struktur data
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 250 obs. of  2 variables:
##  $ income: num  233 250 204 236 354 192 294 263 199 168 ...
##  $ spend : num  150 187 172 178 163 148 153 173 162 174 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   income = col_double(),
##   ..   spend = col_double()
##   .. )
```

```
summary(df.income) # menampilkan rangkuman data
```

```
##      income          spend
##  Min.   :126.0   Min.   : 88.0
##  1st Qu.:211.2   1st Qu.:140.2
##  Median :243.0   Median :156.0
##  Mean   :246.3   Mean   :152.5
##  3rd Qu.:274.0   3rd Qu.:169.0
##  Max.   :417.0   Max.   :202.0
```

```
# Mengetahui jumlah data kosong
sum(is.na(df.income))
```

```
## [1] 0
```

```
# Visualisasi data
ggplot(df.income, aes(x = income, y = spend, colour = income)) +
  geom_point(size = 2, shape = 22, fill = "NA")
```

Standardisasi data

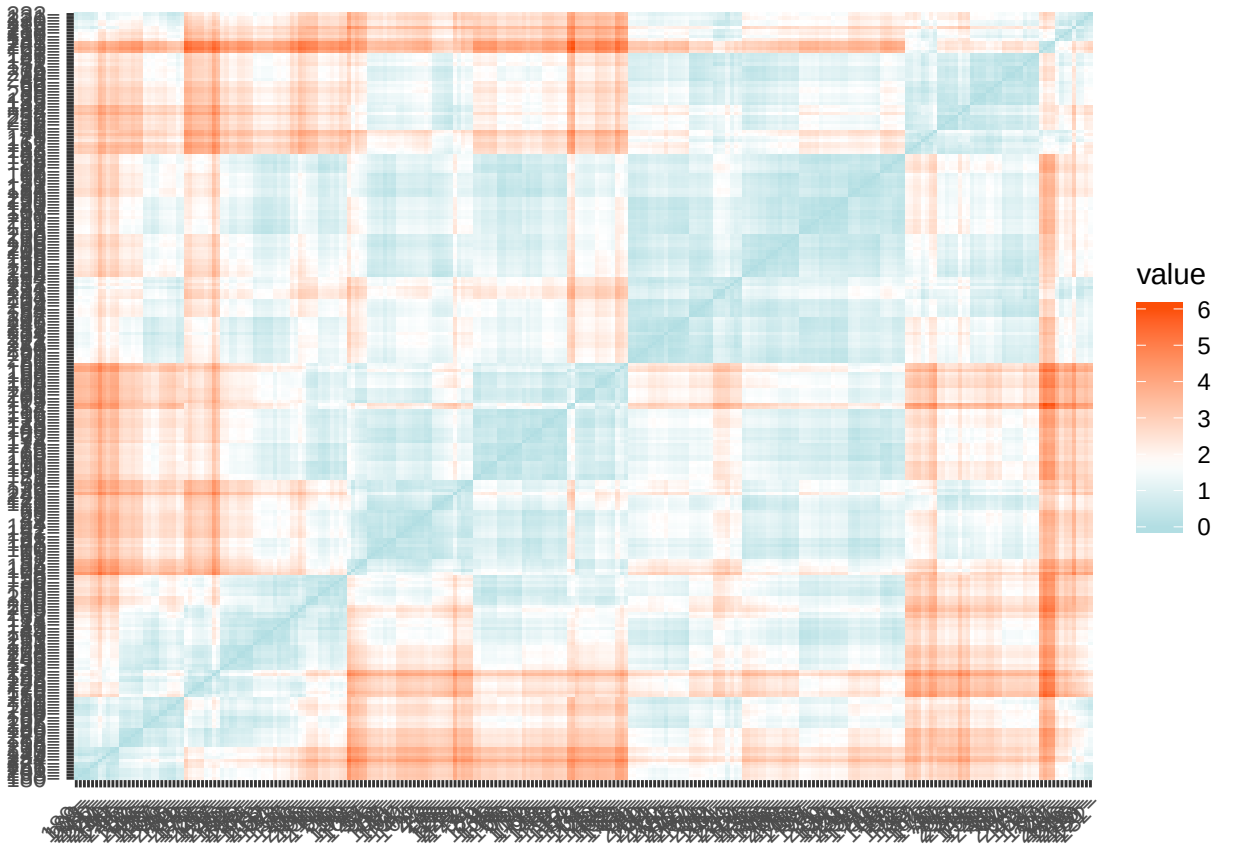
Selanjutnya, untuk membuat semua feature dalam dataset memliki skala yang sama, kita perlu melalukan standarisasi / penyetaraan skala. Standarisasi dapat dilakukan menggunkan fungsi "scale()".

```
# Standardisasi data
incomescaled <- scale(df.income)

# Menampilkan hasil standardisasi data dalam bentuk tabel
df.incomescaled <- data.frame(incomescaled)
View(df.incomescaled)

# Visualisasi data yang telah distandardisasi
ggplot(df.incomescaled, aes(x = income, y = spend, colour = income)) +
  geom_point(size = 2, shape = 22, fill = "NA")
```

Mengukur jarak antar titik

Pemilihan disctance measures penting karena berpengaruh pada hasil clustering. Beberapa distance measures yang umum digunakan yaitu euclidean and manhattan distances.

```
# Menghitung distance
distance <- dist(df.incomescaled, method = "euclidean")

# Menampilkan distance dalam bentuk tabel
df.distance <- data.frame(as.matrix(distance))
view(df.distance)

# Menampilkan distance dalam bentuk heatmap
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



K-Means Clustering

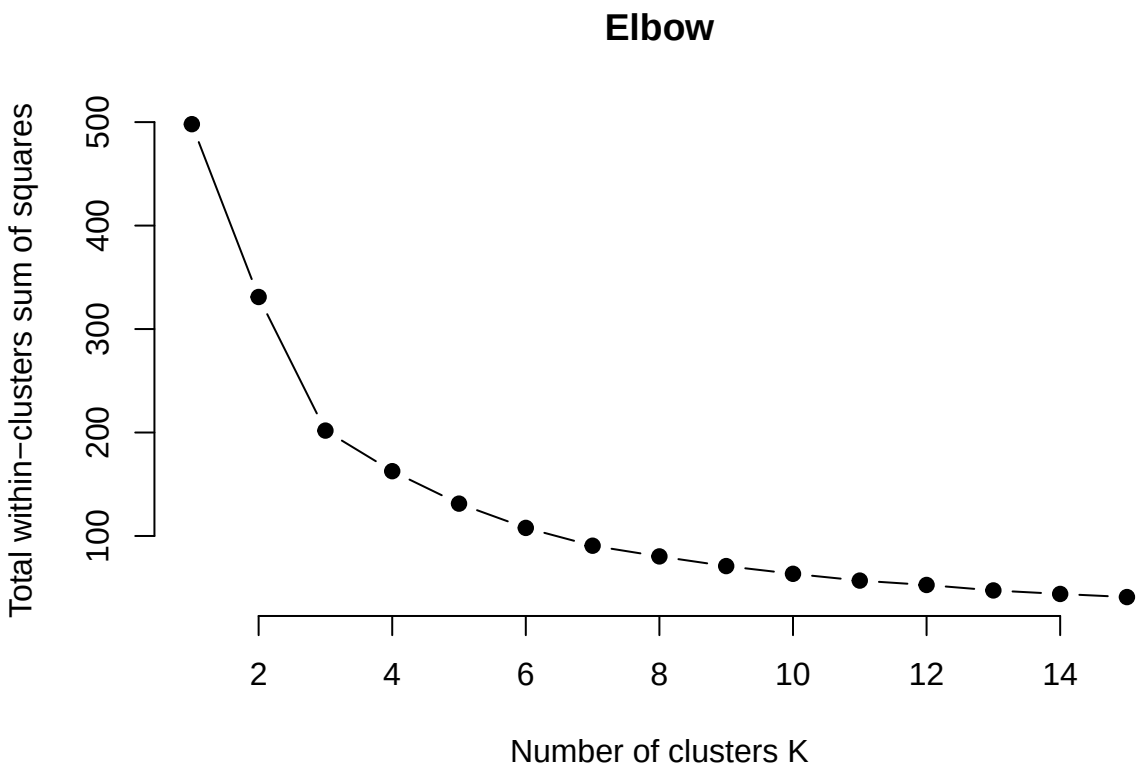
K-means clustering adalah metode pengelompokan yang paling sederhana dan umum digunakan untuk memisahkan dataset ke dalam sejumlah k sub-kelompok.

Menentukan Jumlah K

Jumlah cluster (k) harus ditetapkan sebelum kita memulai algoritma. Jumlah cluster optimum dapat dilakukan dengan menggunakan dua metode yang populer yaitu metode elbow.

```
# Membuat fungsi untuk menghitung within-cluster sum of square
wss <- function(k) {kmeans(df.incomescaled, k, nstart = 25)$tot.withinss}

# Menghitung wss untuk k = 1 sampai k = 15
k.values <- 1:15
wss.values <- map_dbl(k.values, wss)

# Melihat tabel wss untuk k = 1 sampai k = 15
df.wss <- setNames(data.frame(k.values, wss.values), c("k", "wss"))
View(df.wss)

# Visualisasi wss untuk k = 1 sampai k = 15
plot(k.values, wss.values,
  type = "b", pch = 19, frame = FALSE,
  xlab = "Number of clusters K",
  ylab = "Total within-clusters sum of squares",
  main = "Elbow"
```

```
)
```

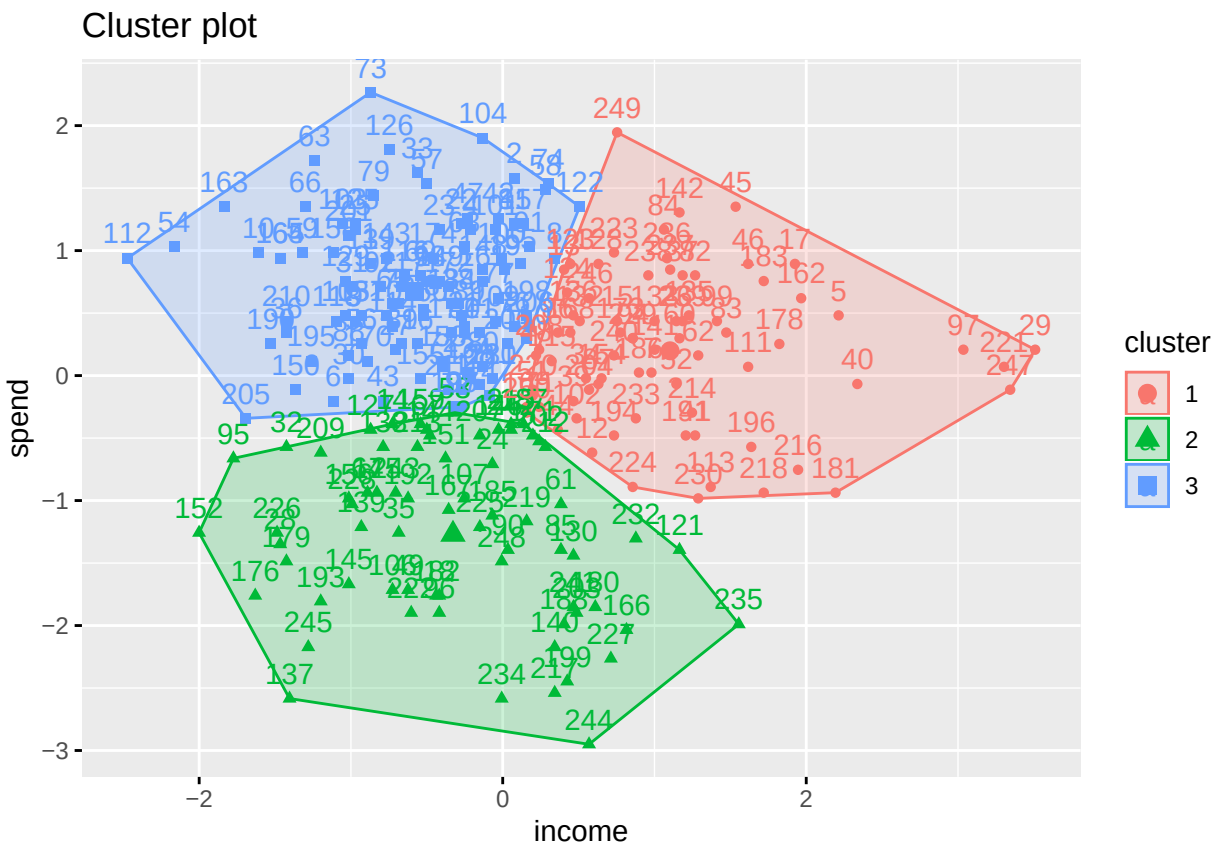


Hasilnya menunjukkan bahwa 3 adalah jumlah kluster optimal karena berada diposisi paling siku.

Membuat Model K-Means

Model K-Means dibuat berdasarkan jumlah kluster optimal yang telah ditentukan.

```
# Membuat model K-Means
km.income <- kmeans(df.incomescaled, centers = 3, nstart = 25)

# Visualisasi cluster
fviz_cluster(km.income, data = df.incomescaled)
```



Hierarchical Clustering

Berbeda dengan k-means, hierarchical clustering tidak membutuhkan penentuan jumlah kluster. Terdapat berbagai cara pengelompokan dalam hierarchical clustering, yang paling populer antara lain: single linkage, complete linkage, average, centroid, dan ward.

Membuat Model Hierarchical Clustering

```
# Membuat model hierarchical clustering

# Hierarchical clustering - single linkage
hc.single <- hclust(distance, method = "single")

# Hierarchical clustering - complete linkage
hc.complete <- hclust(distance, method = "complete")

# Hierarchical clustering - average
hc.average <- hclust(distance, method = "average")

# Hierarchical clustering - centroid
hc.centroid <- hclust(distance, method = "centroid")

# Hierarchical clustering - ward
hc.ward <- hclust(distance, method = "ward.D")
```

Single Linkage, prosedur ini didasarkan pada jarak terkecil. Jika dua obyek terpisah oleh jarak yang pendek

maka kedua obyek tersebut akan digabung menjadi satu cluster daan demikian saterusnya.

Complete Linkage, berlawanan dengan Single Linkage prosedur ini pengelompokkannya berdasarkan jarak terjauh.

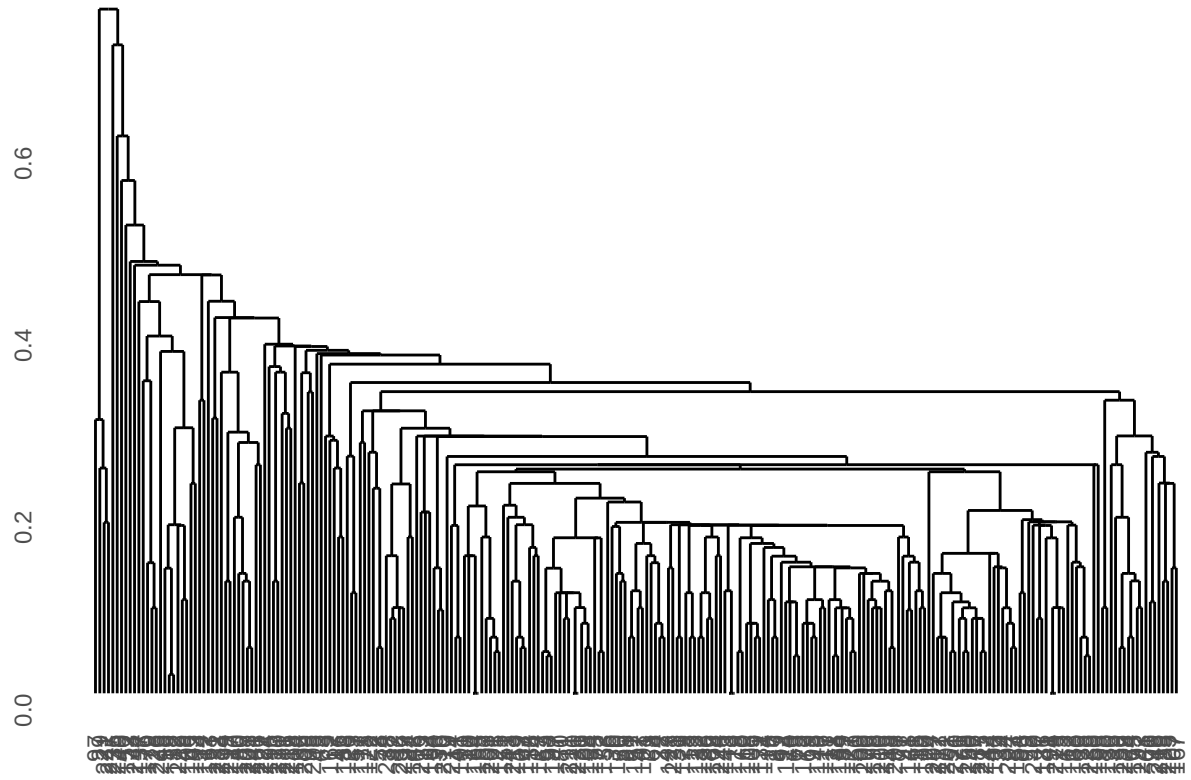
Average Linkage, prosedure ini hampir sama dengan Single Linkage maupun Complete Linkage, namun kriteria yang digunakan adalah rata-rata jarak seluruh individu dalam suatu cluster dengan jarak seluruh individu dalam cluster yang lain.

Centroid Method, jarak antara dua cluster dalam metode ini berdasarkan jarak centroid dua cluster yang bersangkutan.

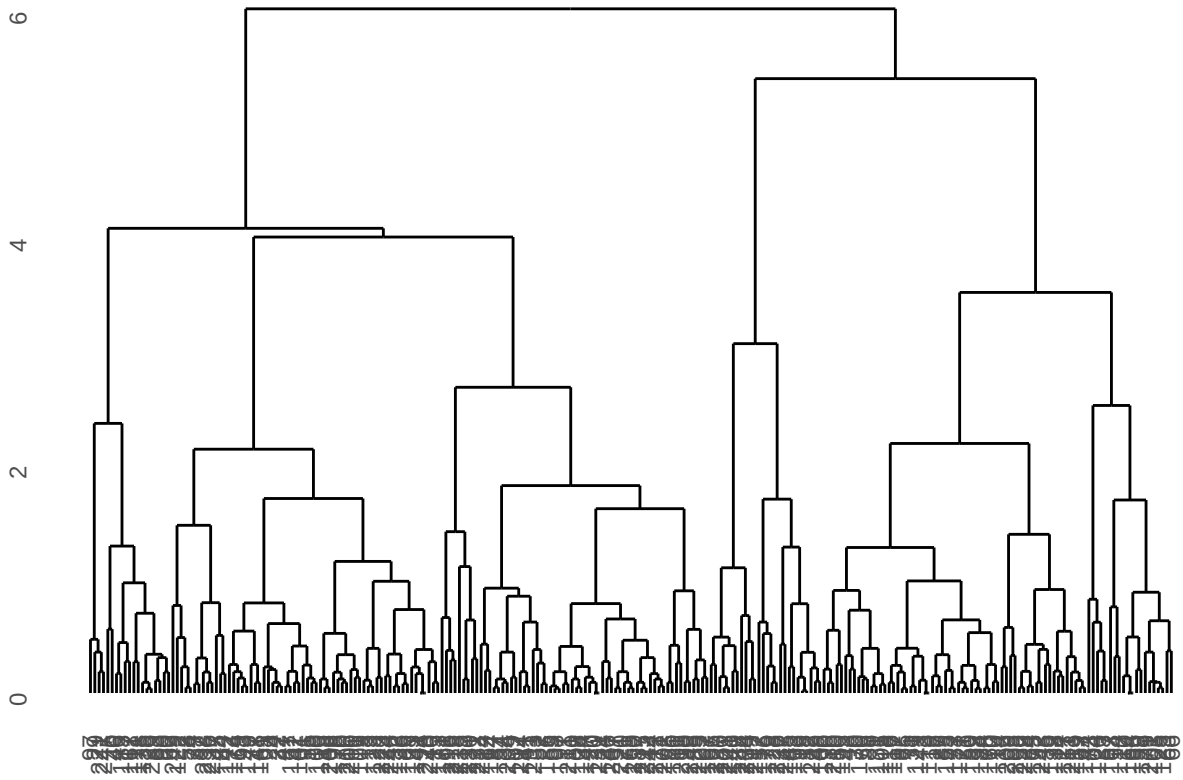
Ward's Method, jarak antara dua cluster dalam metode ini berdasarkan total sum of square dua cluster pada masing-masing variabel.

Visualisasi Model Hierarchical Clustering

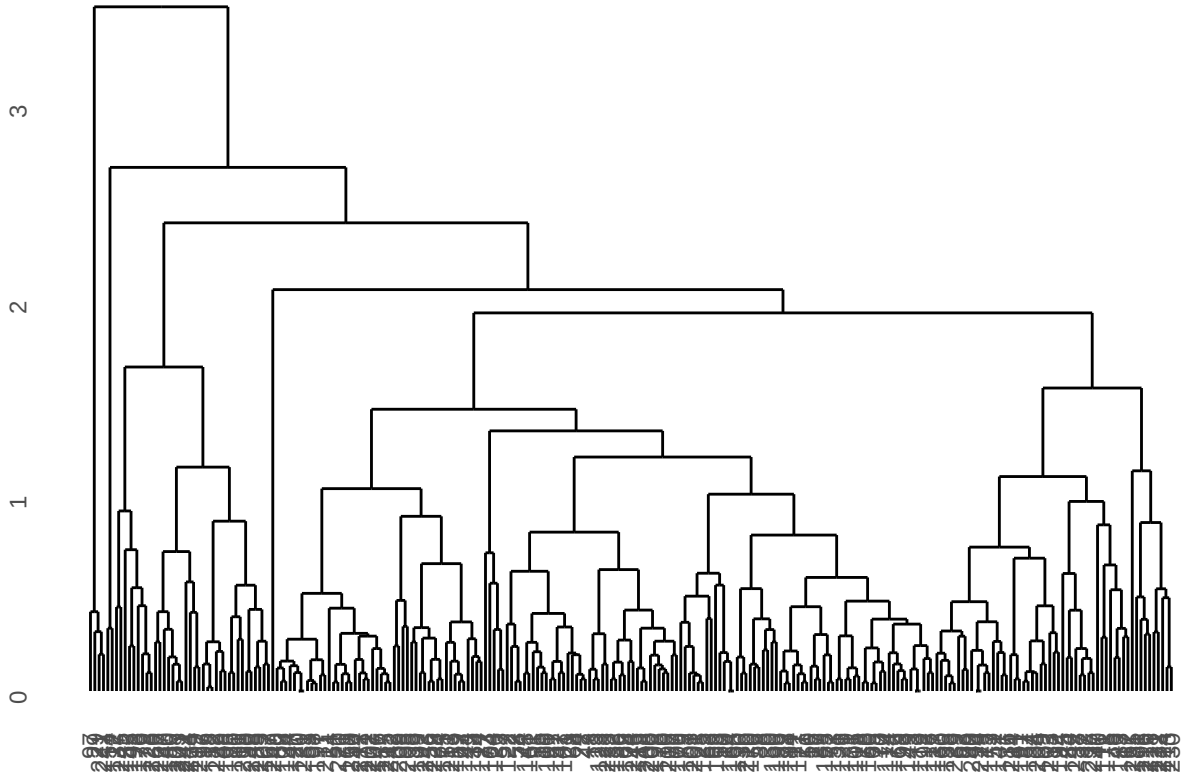
```
# visualisasi model hierarchical clustering

# Hierarchical clustering - single linkage
plot.single <- ggdendrogram(hc.single, rotate = FALSE)
plot.single
```



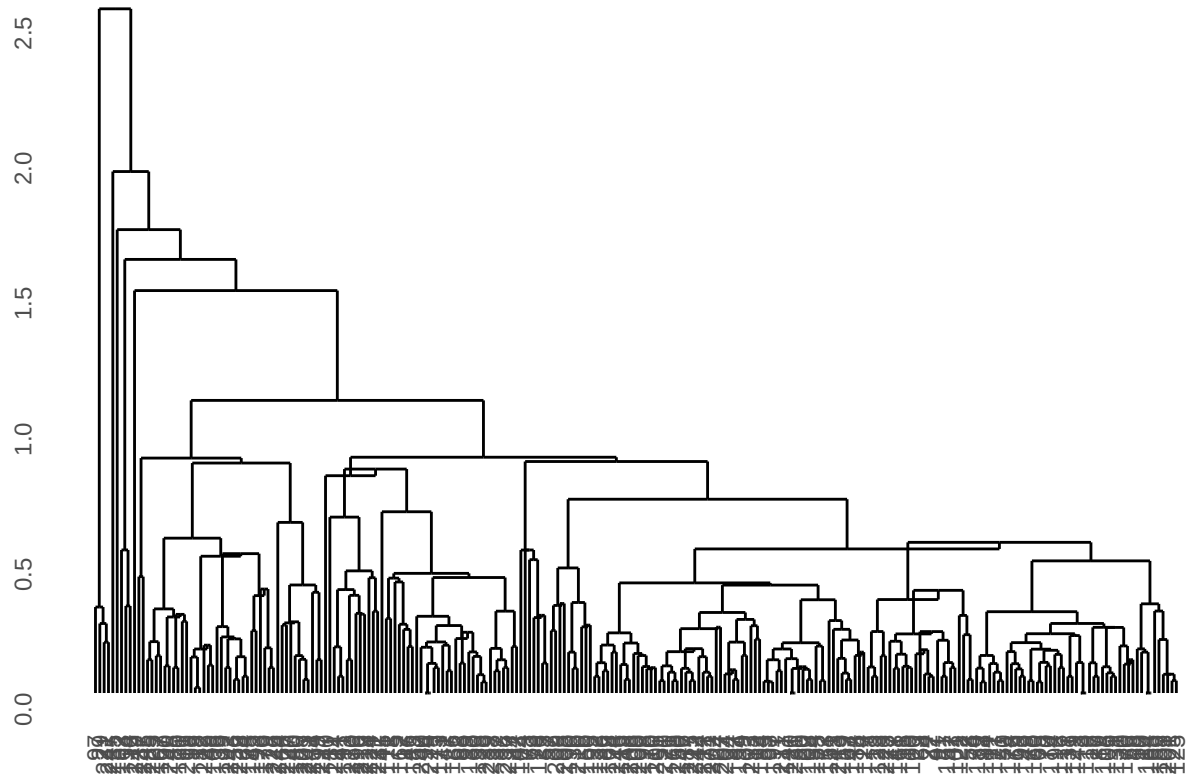

```
# Hierarchical clustering - complete linkage
plot.complete <- ggdendrogram(hc.complete, rotate = FALSE)
plot.complete
```



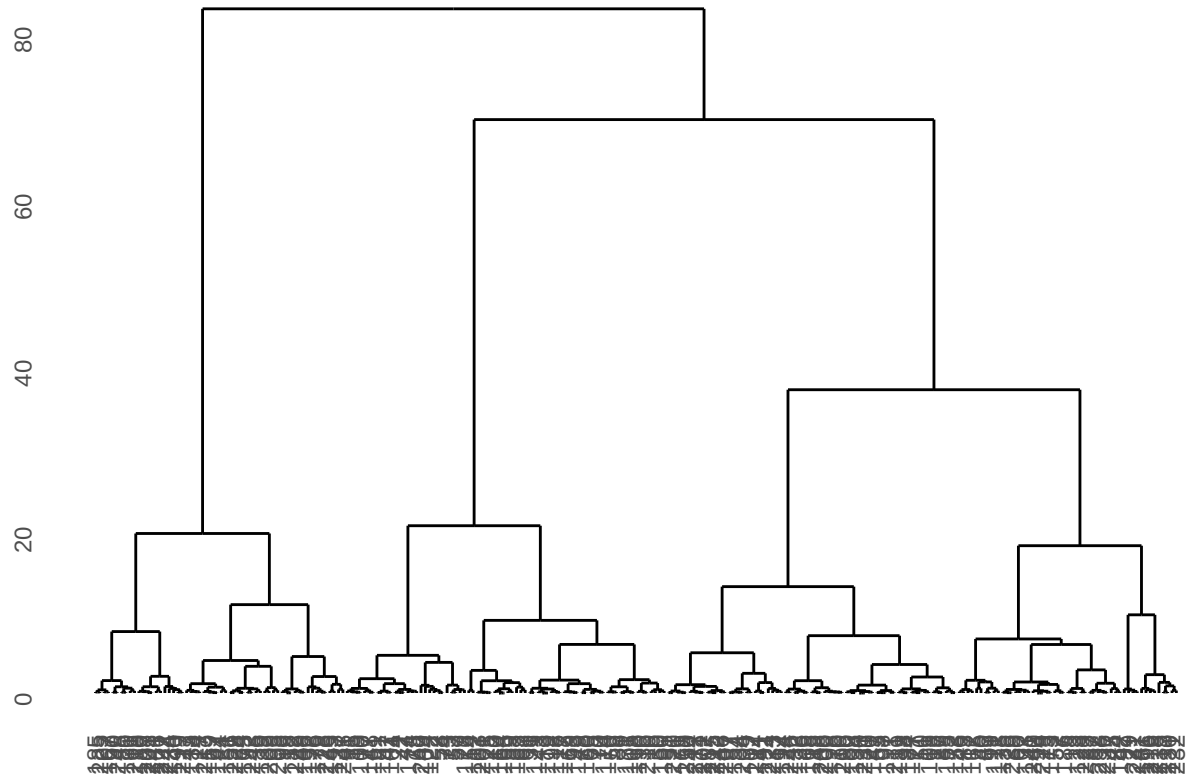

```
# Hierarchical clustering - average
plot.average <- ggdendrogram(hc.average, rotate = FALSE)
plot.average
```




```
# Hierarchical clustering - centroid
plot.centroid <- ggdendrogram(hc.centroid, rotate = FALSE)
plot.centroid
```




```
# Hierarchical clustering - ward
plot.ward <- ggdendrogram(hc.ward, rotate = FALSE)
plot.ward
```




```
# Show plots
require(gridExtra)
```

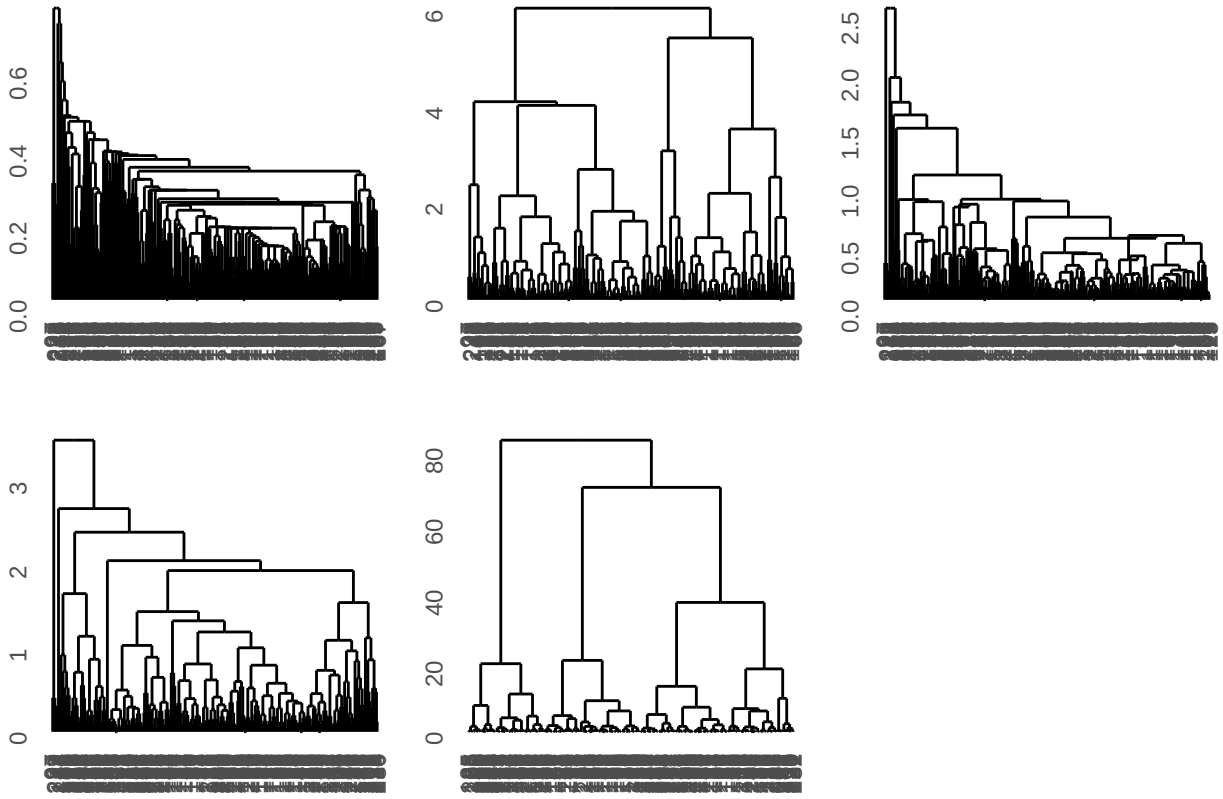
```
## Loading required package: gridExtra
```

```
## Warning: package 'gridExtra' was built under R version 3.6.2
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(plot.single, plot.complete, plot.centroid, plot.average, plot.ward, nrow = 2)
```



Hierarchical Clustering dengan 3 Kluster

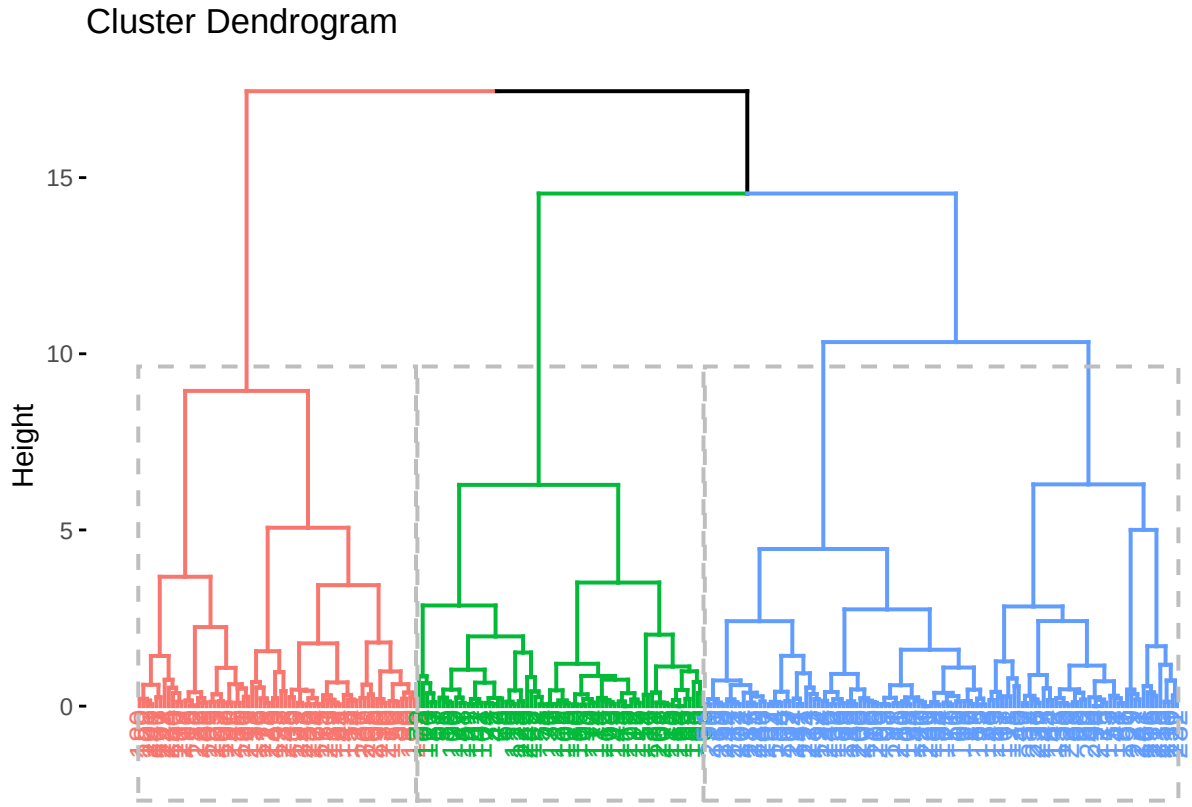
```
# Membuat hierarchical cluster dengan jumlah kluster = 3
clusterincome <- hcut(distance, k = 3, hc_method = "ward.D2")

# Menampilkan pengelompokan
clusterincome$cluster
```

```
##   [1] 1 2 2 2 1 3 1 2 2 2 1 1 2 3 1 1 1 3 2 1 1 2 2 1 2 1 1 3 1 3 2 3 2 1 3
##  [36] 3 1 1 1 1 1 2 3 1 1 1 2 1 3 1 1 1 1 2 2 3 2 2 2 1 3 1 2 3 2 2 3 2 2 2
##  [71] 1 1 2 2 2 1 2 2 2 1 2 1 1 1 3 1 1 3 1 3 2 2 2 1 3 3 1 1 1 2 2 1 2 2 2
## [106] 3 3 1 1 1 1 2 1 2 1 2 2 2 2 1 1 2 2 1 2 2 3 2 2 3 2 1 1 1 1 1 3 3 3 3
## [141] 1 1 2 3 3 1 1 2 2 3 3 3 1 1 1 3 2 2 3 2 2 1 2 2 2 3 3 1 1 3 1 2 1 2 3
## [176] 3 1 1 3 3 1 3 1 1 3 1 1 3 1 3 1 3 3 1 3 1 1 1 3 1 2 1 3 1 3 1 1 1 3 3
## [211] 1 1 3 1 1 1 3 1 3 1 1 3 2 1 3 3 3 3 1 1 1 1 1 3 1 1 1 1 1 1 3 1 3 3 3
## [246] 1 1 3 2 1
```

```
# Menampilkan ukuran kluster
clusterincome$size
```

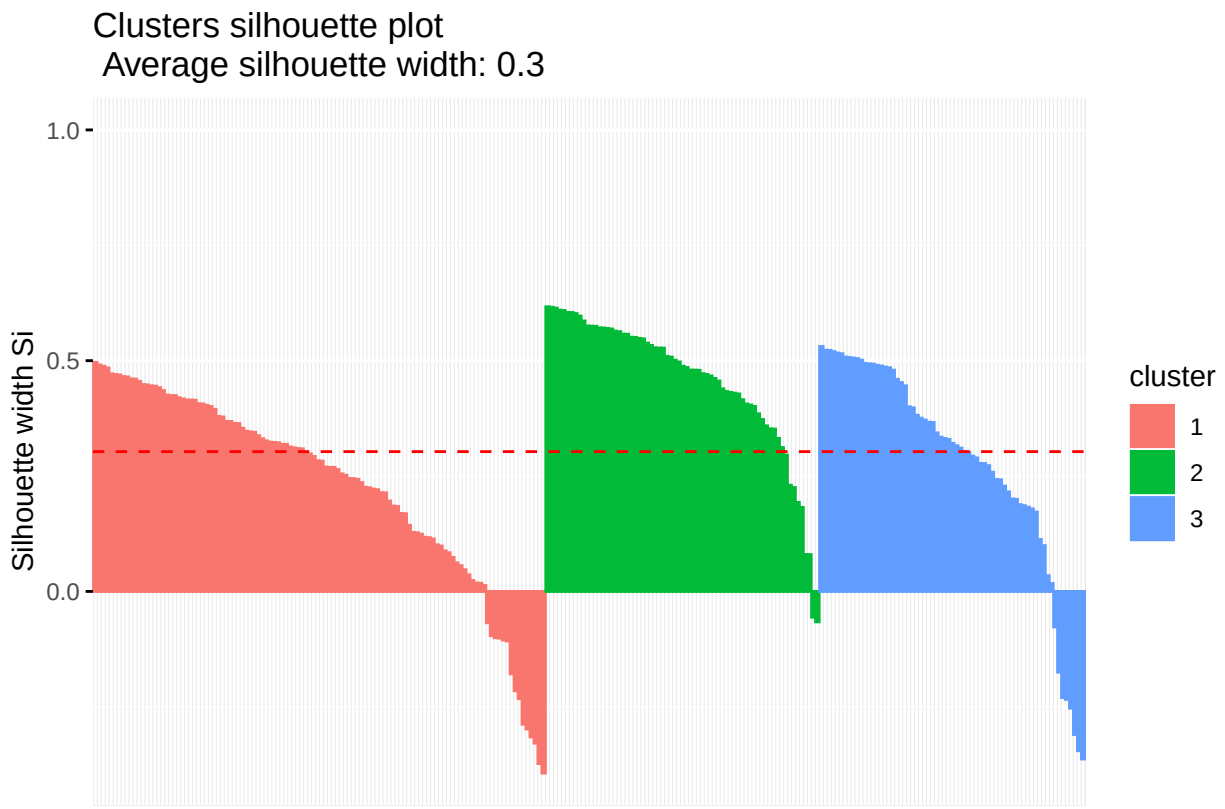
```
## [1] 114  69  67
```

```
# Visualisasi dendogram
fviz_dend(clusterincome, rect = TRUE)
```

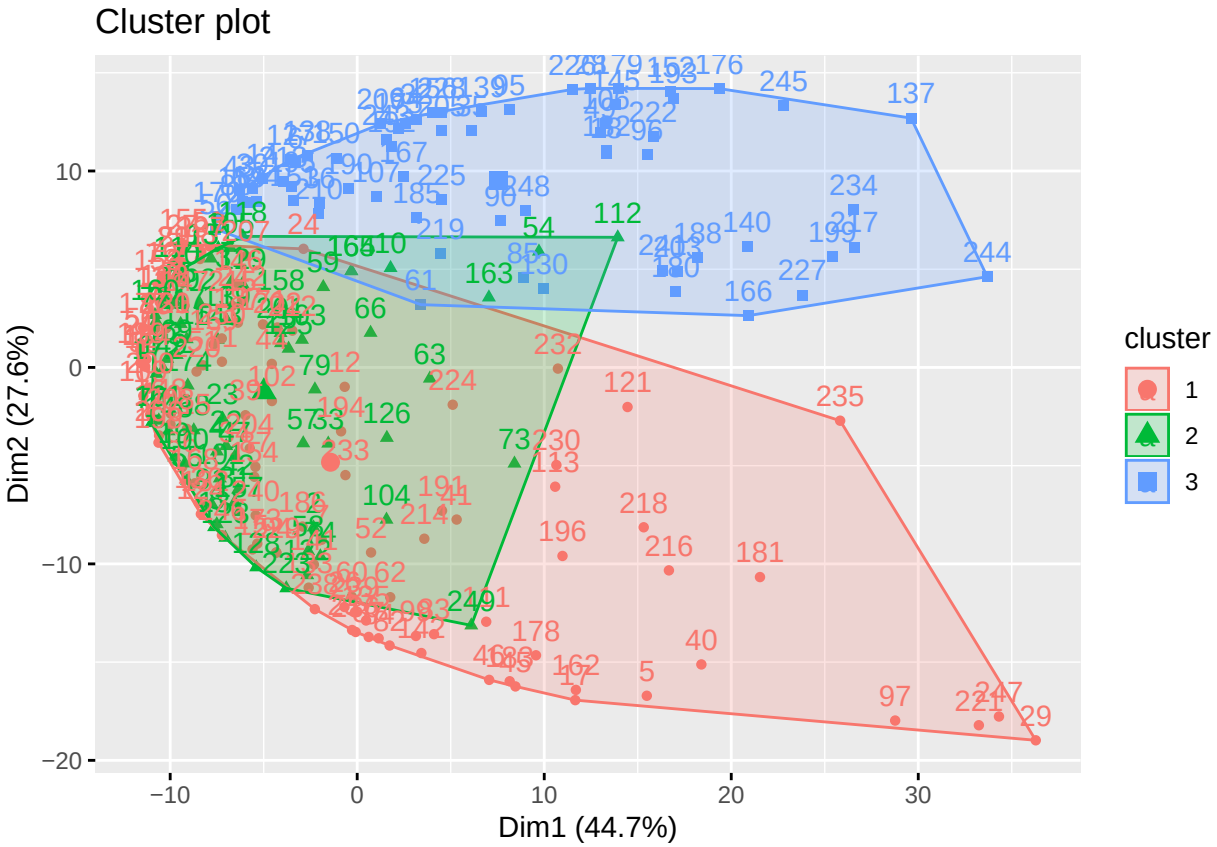



```
# Visulasisasi siluet
fviz_silhouette(clusterincome)
```

```
##   cluster size ave.sil.width
## 1       1  114          0.23
## 2       2   69          0.45
## 3       3   67          0.28
```




```
# Visualisasi sebaran kluster
fviz_cluster(clusterincome, data = distance)
```


Cluster plot
Dim2 (27.6%)
Dim1 (44.7%)
cluster
1
2
3