Nama anggota:

Erick Marcellino Pranata – 210711155

Jacklyn Fionadewi Suseno – 210711210

Elluy Gabriel Panambe – 210711306

Alfa Nada Yulaswara – 210711378

UTS Kapita Selekta

Step 1:

```
data_ecommerce <- read.csv("dataset_UTS.csv")
```

mengambil data dari dataset_UTS.csv

Step 2:

```
library(tidyverse)
ecom <- mutate(data_ecommerce[, c(-6)],
               PurchaseDate = as.Date(PurchaseDate, format = '%m/%d/%Y'),
               TransactionAmount = as.numeric(TransactionAmount))

head(ecom)
summary(ecom)

table(complete.cases(ecom))
complete.cases(ecom)
```

Pertama, kita cek struktur dari dataset yang kita gunakan menggunank *str(data_ecommerce)* yang hasilnya sebagai berikut:

```
>   str(data_ecommerce)  # Check structure of the dataset
'data.frame':   1000 obs. of  6 variables:
 $ CustomerID        : int  4608 6911 4608 2559 9409 8483 8814 5670 8993 3519 ...
 $ PurchaseDate      : chr  "11/04/2023" "11/04/2023" "11/04/2023" "11/04/2023" ...
 $ TransactionAmount : chr  "433.33" "272.56" "?" "623.35" ...
 $ ProductInformation: chr  "Product B'" "Product B'" "Product C'" "'Product A'" ...
 $ OrderID           : int  826847 963918 112426 139726 691194 691194 239145 340062 176819 890075 ...
 $ Location          : chr  "'New York'" "Tokyo" "'New York'" "London"
```

Dari sini kita bisa mengetahui bahwa customerID dan OrderID ber typedata integer dan lainya string

Kumudian kita gunakan library tidyverse yang mencakup berbagai paket yang sering digunakan dalam analisis data.

Kemudian kita buang data yang tidak dibutuhkan untuk RFM (yaitu data lokasi di kolom 6), kemudian mengubah typedata tanggal yang tadinya string diformat menjadi date dengan format yang sesuai ('%m/%d/%Y'), transactionAmount juga perlu diubah menjadi numeric (integer) supaya lebih mudah dihitung dan diurutkan untuk kebutuhan RFM

Kemudian kita cek hasilnya dan ringkasanya mengguanakn summary:

```
>   head(ecom)
  CustomerID PurchaseDate TransactionAmount ProductInformation OrderID
1       4608   2023-11-04            433.33         Product B'  826847
2       6911   2023-11-04            272.56         Product B'  963918
3       4608   2023-11-04                NA         Product C'  112426
4       2559   2023-11-04            623.35        'Product A'  139726
5       9409   2023-11-04            839.56        'Product A'  691194
6       8483   2023-11-04            373.23        'Product C'  691194
>   summary(ecom)  # Summary statistics
   CustomerID    PurchaseDate        TransactionAmount ProductInformation    OrderID
 Min.   :1011   Min.   :2023-01-05   Min.   : 12.13    Length:1000        Min.   :100096
 1st Qu.:3266   1st Qu.:2023-04-05   1st Qu.:257.12    Class :character   1st Qu.:313858
 Median :5520   Median :2023-06-05   Median :521.43    Mode  :character   Median :564804
 Mean   :5545   Mean   :2023-06-11   Mean   :512.19                       Mean   :554568
 3rd Qu.:7808   3rd Qu.:2023-09-06   3rd Qu.:760.59                       3rd Qu.:783052
 Max.   :9991   Max.   :2023-12-05   Max.   :999.44                       Max.   :999695
                NA's   :622          NA's   :48
```

Kemudian kita cek juga apakah kelengkapan semua kasus/data

```
>   table(complete.cases(ecom))

FALSE  TRUE
  642   358
```

Dari 1000 data yang ada, terdapat 358 data yang lengkap

Step 3:

```r
ecom$TransactionAmount[is.na(ecom$TransactionAmount)] <- median(ecom$TransactionAmount, na.rm = TRUE)
ecom_new=na.omit(ecom)

table(complete.cases(ecom_new))
summary(ecom_new)

ecom_new = ecom_new %>%
  filter(TransactionAmount > 0)

summary(ecom_new)
str(ecom_new)

library(VIM)
aggr(ecom_new, numbers=TRUE, prop=FALSE)
marginplot(ecom_new[c("PurchaseDate","TransactionAmount")],pch=c(18),col=c("blue","red"))

ecom_new %>%
  arrange(desc(TransactionAmount)) %>%
  tail(10)
```

Pertama tama kita ganti nilai yang hilang di kolom TransactionAmount dengan nilai median nya

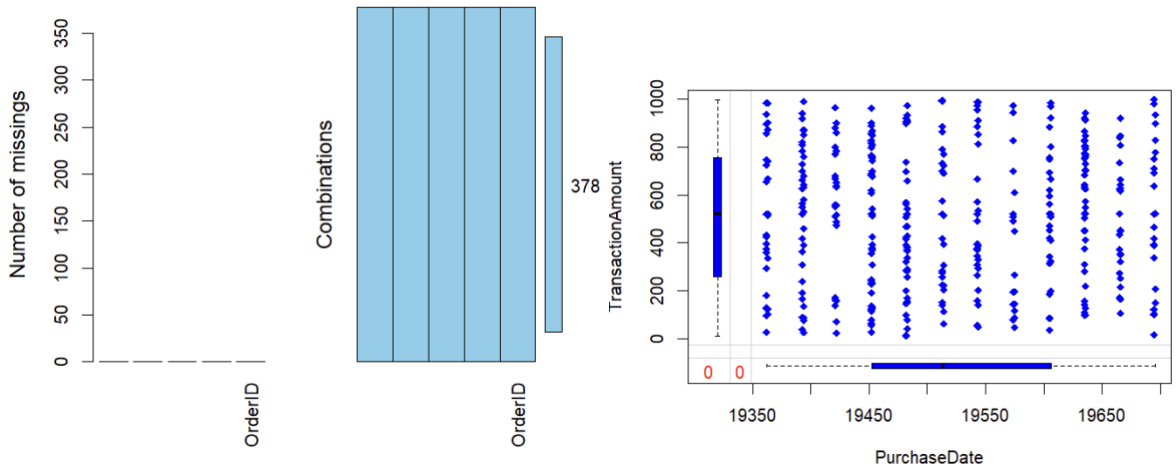
Kemudian kita hapus baris yang hilang, data yang bersih kita simpan di variable baru ecom_new

Hasil dari pembersihannya :

```
TRUE
 378
>   summary(ecom_new)
   CustomerID    PurchaseDate        TransactionAmount ProductInformation    OrderID
 Min.   :1011   Min.   :2023-01-05   Min.   : 12.13    Length:378         Min.   :100205
 1st Qu.:3473   1st Qu.:2023-04-05   1st Qu.:257.43    Class :character   1st Qu.:324083
 Median :5712   Median :2023-06-05   Median :521.43    Mode  :character   Median :566726
 Mean   :5617   Mean   :2023-06-11   Mean   :506.35                       Mean   :562098
 3rd Qu.:7730   3rd Qu.:2023-09-06   3rd Qu.:753.11                       3rd Qu.:787679
 Max.   :9991   Max.   :2023-12-05   Max.   :997.19                       Max.   :999138
>   str(ecom_new)
'data.frame':   378 obs. of  5 variables:
 $ CustomerID       : int  4608 6911 4608 2559 9409 8483 8814 5670 8993 3519 ...
 $ PurchaseDate     : Date, format: "2023-11-04" "2023-11-04" "2023-11-04" "2023-11-04" ...
 $ TransactionAmount : num  433 273 521 623 840 ...
 $ ProductInformation: chr  "Product B'" "Product B'" "Product C'" "'Product A'" ...
 $ OrderID          : int  826847 963918 112426 139726 691194 691194 239145 340062 176819 890075 ...
 - attr(*, "na.action")= 'omit' Named int [1:622] 12 17 28 29 30 31 32 33 34 35 ...
  ..- attr(*, "names")= chr [1:622] "12" "17" "28" "29" ...
```

Kemudian kita gunakan library VIM untuk melakukan analisis lebih lanjut terkait keberadaan data yang hilang menggunakan fungsi aggr().

Kemudian kita buat visualisasi nya menggunakan aggr dan marginplot




Kemudian kita tampilkan 10 data dengan nilai TransactionAmount tertinggi menggunakan fungsi arrange(desc(TransactionAmount)) dan tail(10):

```
+     tail(10)
    CustomerID PurchaseDate TransactionAmount ProductInformation OrderID
369       3926   2023-05-06             40.96        'Product A'  608947
370       7209   2023-02-05             38.41        'Product C'  290672
371       6299   2023-09-05             34.66        'Product D'  718990
372       9331   2023-04-05             28.05        'Product C'  289019
373       8094   2023-01-05             26.72        'Product A'  660814
374       8780   2023-02-06             26.47        'Product C'  806537
375       7251   2023-03-06             22.49        'Product B'  503489
376       8993   2023-12-04             16.55        'Product D'  176819
377       8821   2023-05-05             14.62        'Product B'  866151
378       3697   2023-05-05             12.13        'Product D'  378085
```

Step 4:

```r
library(rfm)

analysis_date <- max(ecom_new$PurchaseDate) + 1

rfm_data <- ecom_new %>%
  group_by(CustomerID) %>%
  summarise(
    Recency = as.numeric(analysis_date - max(as.Date(PurchaseDate, "%m/%d/%Y"))),
    Frequency = n(),
    Monetary = sum(TransactionAmount)
  )
```

Untuk analisis data kita gunakan library RFM

Untuk anlisis date nya kita asumsikan 1 hari setelah transaksi terakhir

Untuk menghitung rfm nya kita gunakan

- Recency = as.numeric(analysis_date - max(as.Date(PurchaseDate, "%m/%d/%Y"))): Ini adalah bagian yang menghitung nilai Recency untuk setiap pelanggan. Nilai Recencynya dihitung sebagai selisih antara analysis_date dengan tanggal pembelian terbaru (max(as.Date(PurchaseDate, "%m/%d/%Y"))) dari pelanggan tersebut.
- Frequency = n(): Nilai Frequency dihitung sebagai jumlah transaksi yang dilakukan oleh pelanggan tersebut. Fungsi n() menghitung jumlah baris dalam setiap kelompok, yang dalam konteks ini adalah jumlah transaksi.
- Monetary = sum(TransactionAmount): Nilai Monetary dihitung sebagai total nilai transaksi yang dilakukan oleh pelanggan tersebut. Fungsi sum() digunakan untuk menjumlahkan nilai dalam kolom TransactionAmount.

```r
rfm_data <- rfm_data %>%
  mutate(
    R_score = ntile(-Recency, 5),
    F_score = ntile(Frequency, 5),
    M_score = ntile(Monetary, 5)
  )

rfm_data <- rfm_data %>%
  mutate(
    R_level = ifelse(R_score %in% 4:5, "High", "Low"),
    F_level = ifelse(F_score %in% 4:5, "High", "Low"),
    M_level = ifelse(M_score %in% 4:5, "High", "Low"),
    Segment = paste(R_level, F_level, M_level, sep = "-")
  )
```

Disini kita memberikan nilai 1-5 untuk setiap nilai RFM nya, Ini dilakukan dengan menggunakan fungsi ntile() dari paket dplyr, yang membagi data menjadi kelompok-kelompok seukuran yang sama dan memberikan nomor kelompok untuk setiap observasi.

```r
rfm_data <- rfm_data %>%
  mutate(
    R_level = ifelse(R_score %in% 4:5, "High", "Low"),
    F_level = ifelse(F_score %in% 4:5, "High", "Low"),
    M_level = ifelse(M_score %in% 4:5, "High", "Low"),
    Segment = paste(R_level, F_level, M_level, sep = "-")
  )
```

Kita menentukan tingkat RFM dengan menggunakan fungsi ifelse(). Jika skor Recency berada di kuartil ke-4 atau ke-5, maka kita menetapkan nilai "High", yang menunjukkan bahwa nilai Recency tersebut tinggi. Jika tidak, kita tetapkan nilai "Low".

```r
rfm_data$Segment <- recode(rfm_data$Segment,
                           "High-High-High" = "Champions",
                           "High-High-Low" = "Loyal Customers",
                           "High-Low-High" = "Potential Loyalist",
                           "High-Low-Low" = "New Customers",
                           "Low-High-High" = "Promising",
                           "Low-High-Low" = "Need Attention",
                           "Low-Low-High" = "About To Sleep",
                           "Low-Low-Low" = "At Risk"
)
```

Kita namai setiap segment tergantung dengan nilai RFM dari yang paling tinggi (High,High,High) sampai paling rendah (Low,Low,Low)

Kemudian kita tampilkan data yang sudah disegmentasikan

```r
>   # Display the distribution of customers across segments
>   segment_counts <- rfm_data %>%
+     group_by(Segment) %>%
+     summarise(Count = n())
>
>   print(segment_counts)
# A tibble: 8 × 2
  Segment            Count
  <chr>              <int>
1 About To Sleep        48
2 At Risk               81
3 Champions             21
4 Loyal Customers       33
5 Need Attention        57
6 New Customers         52
7 Potential Loyalist    42
8 Promising             37
```

Kemudian kita berikan setiap segment karakteristik supaya data nya lebih mudah lagi untuk dibaca

```r
segment_names <- c("Champions", "Loyal Customers", "Potential Loyalists", "New Customers", "Promising", "Needs Attention", "About To Sleep", "At Risk")
segment_characteristics <- c(
  "These customers frequently make purchases, spend a lot, and respond well to promotions. They are your most valuable customers.",
  "These customers frequently make purchases and spend a lot, but do not respond well to promotions.  ",
  "These customers make purchases frequently, do not spend much, but respond well to promotions.",
  "These customers make purchases frequently but do not spend much and do not respond well to promotions.",
  "These customers do not make purchases frequently, but when they do, they spend a lot and respond well to promotions.",
  "These customers do not make purchases frequently, but when they do, they spend a lot. However, they do not respond well to promotions.",
  "These customers rarely make purchases and do not spend much, but they respond well to promotions.",
  "These customers rarely make purchases, do not spend much, and do not respond well to promotions. They might be your least engaged customers."
)

priority_segments <- c("Champions", "Loyal Customers", "Potential Loyalists")
promotional_strategies <- c(
  "Exclusive discounts, loyalty rewards program, early access to new products",
  "Personalized recommendations based on high-spend categories, targeted discounts",
  "Bundle deals including high-spend category items, special offers for mid-spend category"
)

for (i in 1:length(segment_names)) {
  print(paste(segment_names[i], ":", segment_characteristics[i]))
}

for (i in 1:length(priority_segments)) {
  print(paste(priority_segments[i], ":", promotional_strategies[i]))
}
```

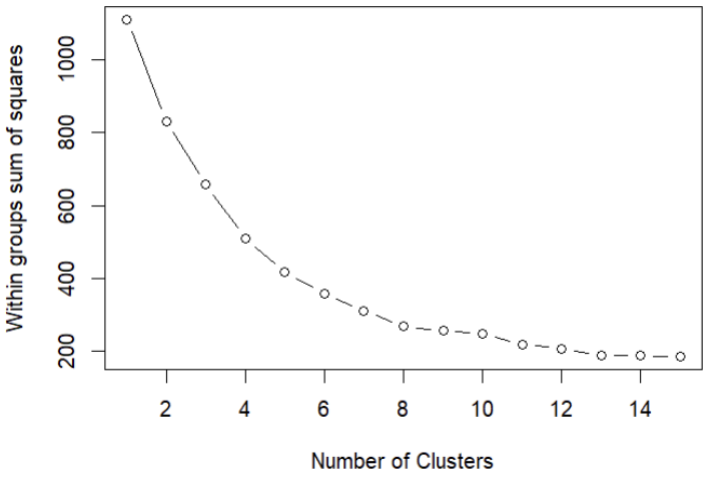
**Metode K-Means dan Elbow**

```r
rfm_cluster <- rfm_data[, c("R_score", "F_score", "M_score")]

scaled_rfm <- scale(rfm_cluster)

wss <- (nrow(scaled_rfm)-1)*sum(apply(scaled_rfm,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(scaled_rfm, centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")

k <- 7
kmeans_model <- kmeans(scaled_rfm, centers = k)

kmeans_data <- rfm_data
kmeans_data$Cluster <- as.factor(kmeans_model$cluster)
```

Pertama tama kita pilih data RFM, yang akan digunakan dimasukan ke variable rfm_cluster

Kemudian kita scaling menggunakan scale() sehingga variabel memiliki mean 0 dan standar deviasi 1.

Kemudian kita menentukan jumlah cluster yang optimal, Kita menghitung within-cluster sum of squares (WSS) untuk jumlah cluster mulai dari 2 hingga 15. Kemudian kita plot kan yang hasilnya:




Bisa dilihat dari grafik elbow nya ada di cluster 7, maka kita gunakan 7 sebagai center nya

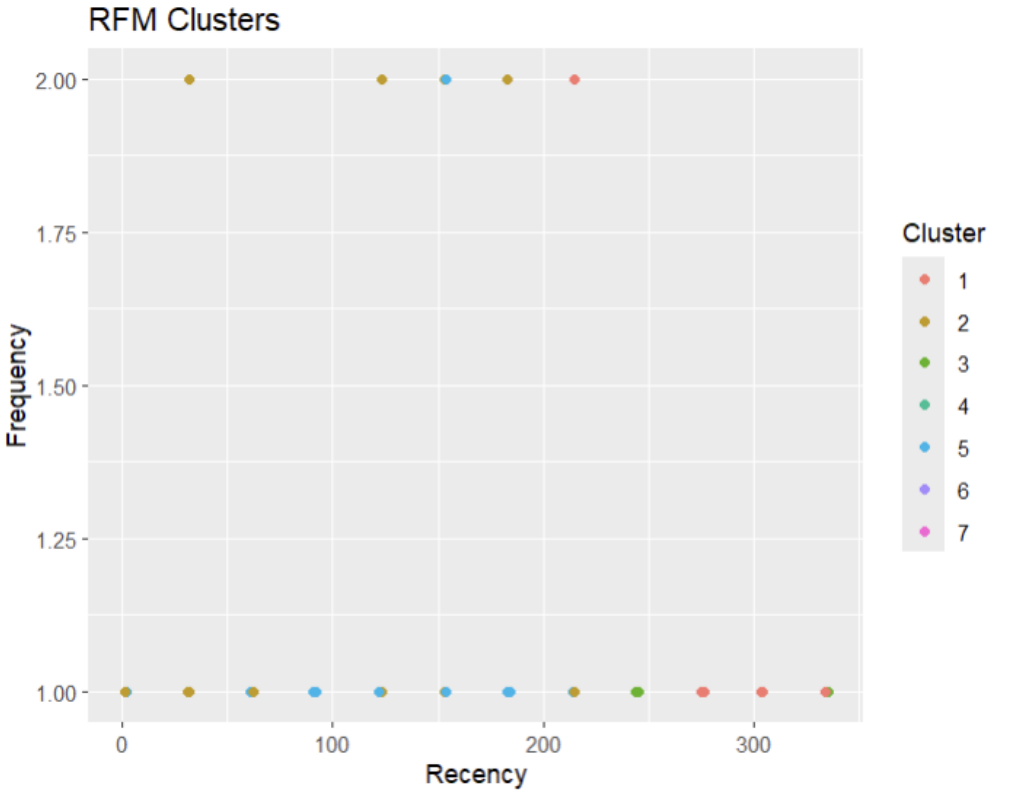
Kemudian kita jalankan algoritma K-Means nya dan dimasukan sebagai K_Means model

```
cluster_summary <- kmeans_data %>%
  group_by(Cluster) %>%
  summarise(
    Count = n(),
    Avg_Recency = mean(Recency),
    Avg_Frequency = mean(Frequency),
    Avg_Monetary = mean(Monetary)
  )
```

print(cluster_summary)

kita ringkas cluster yang kita dapat, yang hasilnya :

```
> print(cluster_summary)
# A tibble: 7 x 5
  Cluster Count Avg_Recency Avg_Frequency Avg_Monetary
  <fct>   <int>       <dbl>          <dbl>        <dbl>
1 1          68        261.           1.01         712.
2 2          31        105.           1.16         908.
3 3          52        257.           1            183.
4 4          46        105.           1            208.
5 5          50        109.           1.02         319.
6 6          61        267.           1            568.
7 7          63         71.4          1            716.
```




```
segment_names <- c("High-Spenders, Recent (41)", "Low-Spenders, Recent (55)", "Medium Spenders, Less Recent (54)", "High-Spenders, Less Recent (43)", "Low-Spenders, Least Recent (73)", "Medium Spenders, Average Recency (50)", "High-Spenders, Dormant (55)"
segment_characteristics <- c(
  "These customers have a low recency (purchase frequently) and a high average order value. They might be loyal customers who regularly make significant purchases.",
  "These customers have low recency (purchase frequently) but a low average order value. They could be new or budget-conscious customers who buy often but in smaller quantities.",
  "These customers have average recency (purchase occasionally) and a moderate average order value. They could be returning customers who buy with some regularity but not as frequently.",
  "These customers have lower recency (purchase less frequently) but the highest average order value. They could be customers who make infrequent, high-value purchases.",
  "These customers have the highest recency (purchase least often) and a low average order value. They could be infrequent, budget-conscious customers.",
  "These customers have average recency and a low average order value. They might be price-sensitive customers who buy occasionally.",
  "These customers have the highest recency (haven't purchased recently) but a high average order value. They could be previously loyal customers who haven't returned in a while but have a history of high spending."
)
```

Memberikan nama dan karakter untuk setiap cluster

```
priority_clusters <- c("Cluster 4", "Cluster 6")
promotional_strategies <- list(
  "Cluster 4" = c(
    "Personalized recommendations: Suggest complementary products based on their purchase history.",
    "Exclusive benefits: Offer them free expedited shipping, extended return windows, or access to a dedicated customer service line.",
    "Early bird offers: Allow them early access to new products or limited-edition items."
  ),
  "Cluster 6" = c(
    "Win-back campaigns: Re-engage them with personalized discounts or special offers based on their past purchase behavior.",
    "Abandoned cart reminders: Remind them about forgotten items in their cart and offer incentives to complete the purchase.",
    "Personalized product recommendations: Recommend items they might be interested in based on their previous purchases and browsing history."
  )
)
```

memprioritaskan Cluster 4 dan Cluster 6 dan mendefinisikan strategi promosi nya

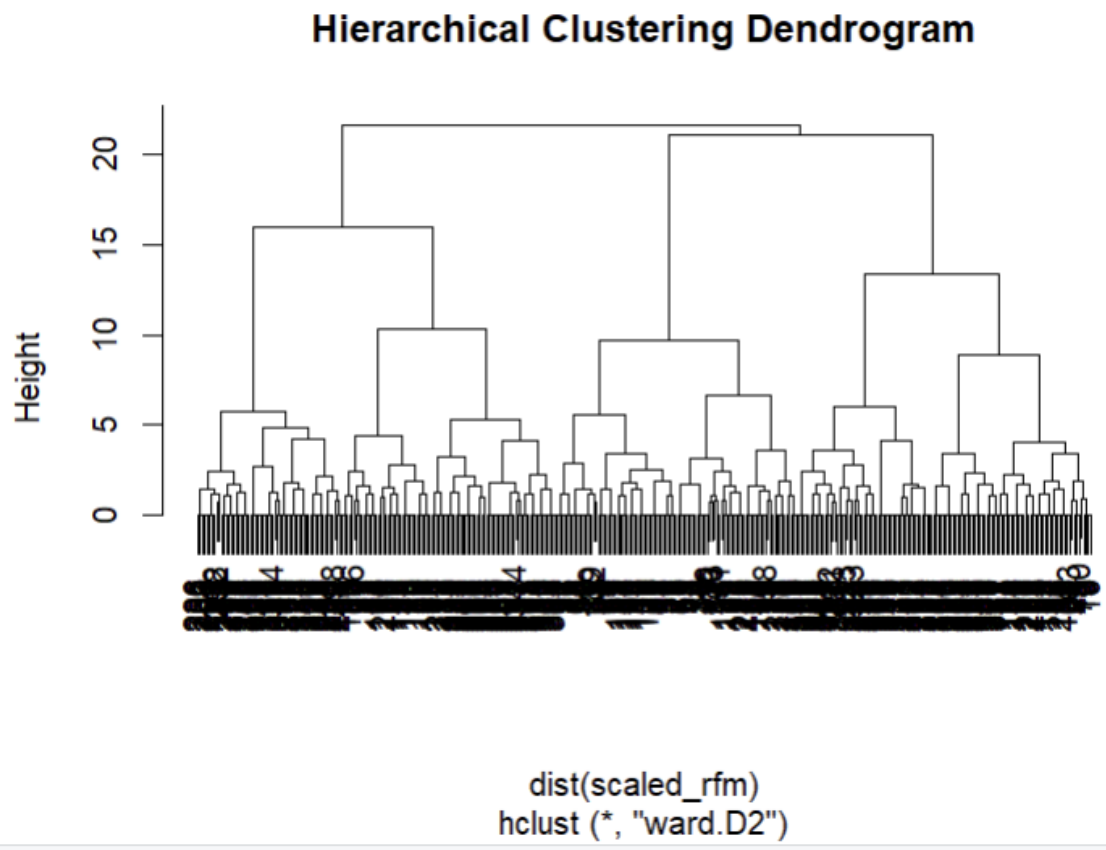
**Menggunakan Hierarchical Clustering**

```
library(dendextend)

rfm_cluster <- rfm_data[, c("R_score", "F_score", "M_score")]

scaled_rfm <- scale(rfm_cluster)

hc <- hclust(dist(scaled_rfm), method = "ward.D2")

plot(hc, main = "Hierarchical Clustering Dendrogram")
rect.hclust(hc, k = 7, border = "black", lty = "dashed")
```

Pertama-tama, kita memilih hanya kolom-kolom R_score, F_score, dan M_score dari data RFM dan menyimpannya dalam variabel rfm_cluster

Kemudian Kita scalingkan variabel RFM menggunakan fungsi scale() sehingga variabel memiliki mean 0 dan standar deviasi 1.

Kemudian kita jalankan algoritma Hierarchical Clustering nya dan dimasukan di variable hc
Hasilnya di plotkan sebagai berikut:



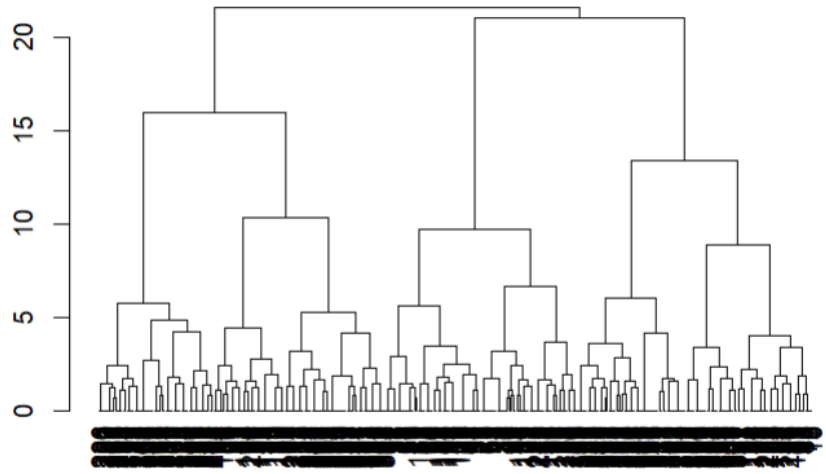

```
k <- 7
clusters <- cutree(hc, k)
```

Kita potong jumlah clusters nya dengan 7 supaya lebih jelas

```
rfm_data$Cluster <- as.factor(clusters)

cluster_summary <- rfm_data %>%
  group_by(Cluster) %>%
  summarise(
    Count = n(),
    Avg_Recency = mean(Recency),
    Avg_Frequency = mean(Frequency),
    Avg_Monetary = mean(Monetary)
  )

print(cluster_summary)

dend <- as.dendrogram(hc)
plot(dend)
```

Sekaligus kita ringkas/summary kan hasilnya:

| | Cluster | Count | Avg_Recency | Avg_Frequency | Avg_Monetary |
|---|---|---|---|---|---|
| | <fct> | <int> | <dbl> | <dbl> | <dbl> |
| 1 | 1 | 60 | 117. | 1.08 | 802. |
| 2 | 2 | 50 | 56.9 | 1 | 688. |
| 3 | 3 | 66 | 266. | 1 | 189. |
| 4 | 4 | 50 | 119. | 1 | 346. |
| 5 | 5 | 37 | 271. | 1 | 713. |
| 6 | 6 | 52 | 287. | 1.02 | 726. |
| 7 | 7 | 56 | 127. | 1.02 | 268. |




Dari cluster yang ada kita berikan nama dan karakteristik:

| Nama | Karakter |
|---|---|
| **High-Spenders, Recent (41)** | Pelanggan ini memiliki tingkat keterkinian yang rendah (sering membeli) dan nilai pesanan rata-rata yang tinggi. Mereka mungkin pelanggan setia yang rutin melakukan pembelian dalam jumlah besar. |
| **Low-Spenders, Recent (55)** | Pelanggan ini memiliki keterkinian yang rendah (sering membeli) tetapi nilai pesanan rata-rata rendah. Mereka bisa jadi pelanggan baru atau pelanggan dengan anggaran terbatas yang sering membeli tetapi dalam jumlah lebih kecil. |
| **Medium Spenders, Less Recent (54)** | Pelanggan ini memiliki keterkinian rata-rata (membeli sesekali) dan nilai pesanan rata-rata sedang. Mereka bisa jadi adalah pelanggan kembali yang membeli secara teratur tetapi tidak sesering itu |
| **High-Spenders, Less Recent (43)** | Pelanggan ini memiliki keterkinian yang lebih rendah (pembelian lebih jarang) tetapi nilai pesanan rata-rata tertinggi. Mereka bisa jadi pelanggan yang jarang melakukan pembelian bernilai tinggi. |
| **Low-Spenders, Least Recent (73)** | Pelanggan ini memiliki keterkinian tertinggi (pembelian paling jarang) dan nilai pesanan rata-rata yang rendah. Mereka mungkin merupakan pelanggan yang jarang dan sadar anggaran |

| | |
|---|---|
| **Medium Spenders, Average Recency (50)** | Pelanggan ini memiliki keterkinian rata-rata dan nilai pesanan rata-rata yang rendah. Mereka mungkin pelanggan sensitif terhadap harga yang sesekali membeli. |
| **High-Spenders, Dormant (55)** | elanggan ini memiliki keterkinian tertinggi (belum membeli baru-baru ini) namun nilai pesanan rata-rata tinggi. Mereka sebelumnya bisa menjadi pelanggan setia yang sudah lama tidak kembali namun memiliki sejarah pengeluaran yang tinggi. |

Kami memilih untuk memprioritaskan cluster 1 dan 2, dengan strategi promosi sebagai berikut:

Cluster 1 : mempertimbangkan program loyalitas atau penghargaan untuk pembelian yang sering dilakukan karena mereka sudah memiliki frekuensi pembelian yang tinggi. Hal ini dapat mendorong mereka untuk mempertahankan atau meningkatkan pembelanjaan mereka.

Cluster 2 : pertimbangkan kampanye keterlibatan kembali seperti penawaran eksklusif atau rekomendasi yang dipersonalisasi untuk mendorong mereka melakukan pembelian lebih sering

**Metode Terbaik**

Dari 3 Metode diatas, kami memilih metode Elbow karena pada metode Elbow untuk menentukan jumlah cluster dapat dibilang paling mudah karena metode Elbow memberikan gambaran untuk membantu menentukan jumlah cluster yang optimal sehingga tingkat keakurasian bisa dibilang paling tinggi dalam menentukan jumlah clusternya.

Setelah didapat jumlah clusternya, maka akan menjadi lebih mudah bagi kita untuk membaginya ke berbagai cluster yang ada. Hasilnya karakteristik pelanggan akan lebih mudah untuk digali secara lebih mendalam.