

CLUSTERING

Tugas 2 | PD - A | 2019

KELOMPOK 1

Firin Handayani
Humaira Nur Pradani

05211640000006
05211640000011



PENDAHULUAN

PENGENALAN DATASET	3
INFORMASI ATRIBUT	3

DASAR TEORI

BISNIS RITEL	4
CUSTOMER RELATIONSHIP MANAGEMENT (CRM)	4
ANALISIS RFM.....	4

BAB I : EKSPLORASI DATA

SUMMARIZATION DATA.....	5
KARAKTERISTIK TIAP ATRIBUT	5
DISTINCT VALUE	5
DIMENSI	6
HEAD & TAIL.....	7
SUMMARY	8
DESCRIBE.....	8
DISTRIBUSI KELAS.....	10
VISUALISASI DATA.....	11
TRANSAKSI PER NEGARA.....	11
JUMLAH PELANGGAN UNIK DI TIAP NEGARA	13
PERSEBARAN HARGA BARANG YANG DIJUAL	14
TOTAL PENJUALAN PER BULAN PADA TAHUN 2009-2011	15
KUANTITAS PEMBELIAN TIAP PELANGGAN	16
JUMLAH BARANG YANG TERJUAL.....	17
BOXPLOT HARGA & KUANTITAS BARANG.....	18

BAB II : PRA-PROSES DATA

MENGGABUNGKAN DATA.....	19
DATA DUPLIKAT	19
MISSING VALUE.....	20
MENGHILANGKAN DATA YANG TIDAK DIPERLUKAN	20
FORMAT DATA WAKTU	22
SAMPLING	22
PENANGANAN OUTLIER PERTAMA.....	23
MEMBUAT BENTUK DATA RFM	24
PENANGANAN OUTLIER KEDUA.....	25



DAFTAR ISI

MEMBUAT DATA CLUSTER DAN SCALING	25
BAB III : CLUSTERING	
K-MEANS CLUSTERING	26
METODE HIRARKI	28
Agglomerative Clustering.....	28
Divisive Clustering	32
METODE BERBASIS DENSITAS (DBSCAN)	33
BAB IV : HASIL CLUSTERING	
PERBANDINGAN HASIL CLUSTERING	34
Kmeans.....	34
Hierarchical Clustering (Agglomerative Clustering-Ward).....	36
DBSCAN	37
PERHITUNGAN JUMLAH CLUSTER OPTIMAL.....	39
K-MEANS	39
HIERARCICAL CLUSTERING	40
BAB V : KESIMPULAN DAN SARAN	
ANALISIS CLUSTER TERHADAP CRM.....	41
KESIMPULAN	43
SARAN	43



PENDAHULUAN

Pengenalan Dataset



Dataset yang digunakan adalah “Online Retail” yang merupakan rangkaian data perusahaan multinasional untuk bisnis ritel. Data tersebut merupakan set data transaksional yang terjadi antara tanggal 01/12/2009 s.d. 09/12/2011 untuk bisnis ritel online non-toko di UK. Perusahaan utamanya menjual barang-barang cinderamata. Kebanyakan kastemer dari perusahaan adalah pedagang grosir.

Dataset tersebut memiliki beberapa karakteristik sebagai berikut :

Karakteristik Dataset	Multivariate, Sequential, Time-Series
Karakteristik Atribut	Numeric (Integer & Real), Categorical
Number of Instances:	1067371
Associated Tasks	Clustering
Area	Business

Informasi Atribut

Berikut merupakan atribut-atribut yang ada pada data :

No.	Atribut	Informasi	Type Data	Keterangan
1	InvoiceNo	Nomor faktur	Nominal	Angka integral 6 digit yang ditetapkan secara unik untuk setiap transaksi. Jika kode ini dimulai dengan huruf 'c', ini menunjukkan pembatalan
2	StockCode	Kode produk (item)	Nominal	Nomor integral 5 digit yang ditetapkan secara unik untuk setiap produk yang berbeda
3	Description	Nama produk (item)	Nominal	-
4	Quantity	Kuantitas setiap produk (item) per transaksi	Numerik	-
5	InvoiceDate	Tanggal dan waktu layanan	Numerik	Hari dan waktu ketika setiap transaksi dihasilkan.
6	UnitPrice	Harga satuan	Numerik	Harga produk per unit dalam sterling
7	CustomerID	Nomor pelanggan	Nominal	Nomor integral 5 digit yang ditetapkan secara unik untuk setiap pelanggan.
8	Country	Nama Negara	Nominal	nama negara tempat setiap pelanggan tinggal



DASAR TEORI

BISNIS RITEL

Pengertian bisnis ritel adalah sebuah bisnis yang menjalankan penjualan barang atau jasa secara eceran atau satuan. Dan produknya langsung ditujukan kepada konsumen untuk memenuhi kebutuhan pribadinya bukan sebagai produk yang akan dijual kembali atau diproses sebagai bahan membuat produk lain.

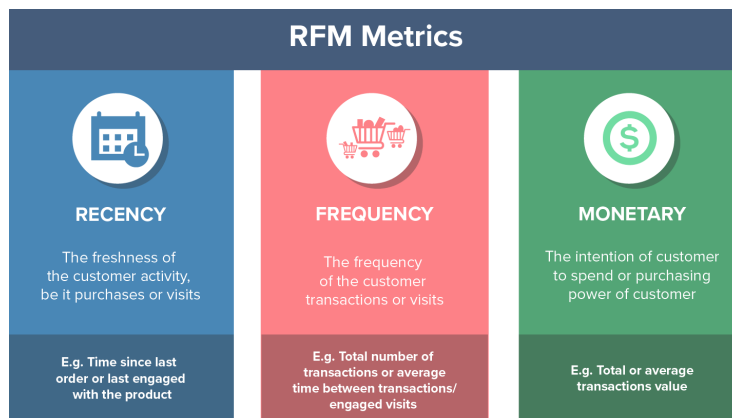
CUSTOMER RELATIONSHIP MANAGEMENT (CRM)

Customer Relationship Management adalah strategi untuk mengelola hubungan dan interaksi organisasi dengan pelanggan dan pelanggan potensial. Sistem CRM membantu perusahaan tetap terhubung dengan pelanggan, merampingkan proses, dan meningkatkan profitabilitas.

Ketika orang berbicara tentang CRM, mereka biasanya mengacu pada sistem CRM, alat yang digunakan untuk manajemen kontak, manajemen penjualan, produktivitas, dan banyak lagi. Tujuan dari sistem CRM sederhana: Meningkatkan hubungan bisnis.

ANALISIS RFM

Analisis RFM (RFM Analysis) adalah metode empiris, yang sangat bergantung pada data dari analisis web, manajemen hubungan pelanggan atau transaksi. Sementara banyak pendekatan pemasaran didasarkan pada karakteristik demografi, analisis RFM melengkapi arah strategis kampanye dengan komponen perilaku. Untuk tujuan ini, perilaku pembelian di masa lalu diperiksa secara lebih rinci:



- **R – Recency – Keterkinian:** Keterkinian pembelian adalah alat penting untuk mengidentifikasi pelanggan yang telah membeli sesuatu baru-baru ini. Pelanggan yang membeli belum lama ini lebih cenderung bereaksi terhadap penawaran baru daripada pelanggan yang pembelannya terjadi sejak lama.
- **F – Frequency – Frekuensi:** Frekuensi pembelian muncul setelah keterkinian. Jika pelanggan membeli lebih sering, kemungkinan respons positif lebih tinggi daripada pelanggan yang jarang membeli sesuatu.
- **M – Monetary Value – Nilai Uang:** Omset pembelian atau nilai moneter mengacu pada semua pembelian yang dilakukan oleh pelanggan. Pelanggan yang menghabiskan lebih banyak uang untuk pembelian lebih cenderung menanggapi penawaran daripada pelanggan yang telah menghabiskan jumlah yang lebih kecil.



SUMMARIZATION DATA KARAKTERISTIK TIAP ATRIBUT

Dalam melakukan eskplorasi data, hal yang pertama dilakukan yaitu dengan mengetahui karakteristik atribut dalam setiap data yaitu melihat struktur data. Penggunaan fungsi `str(nama_data_frame)` bertujuan untuk melihat tipe dan struktur dari setiap data frame. Selain itu, fungsi ini akan menampilkan jumlah baris, nama variabel, tipe variabel, dan sebagian baris pertama dari data. Di bawah ini merupakan tampilan penggunaan syntax dan hasil dari struktur data.

Syntax
<pre>#struktur data str(data_gabung_awal)</pre>
Hasil
<pre>Classes 'tbl_df', 'tbl' and 'data.frame': 1067371 obs. of 8 variables: \$ Invoice : chr "489434" "489434" "489434" "489434" ... \$ StockCode : chr "85048" "79323P" "79323w" "22041" ... \$ Description: chr "15CM CHRISTMAS GLASS BALL 20 LIGHTS" "PINK CHERRY LIGHTS" "WHITE CHERRY LIGHTS" "RECORD FRAME 7\" SINGLE SIZE" ... \$ Quantity : num 12 12 12 48 24 24 24 10 12 12 ... \$ InvoiceDate: POSIXct, format: "2009-12-01 07:45:00" "2009-12-01 07:45:00" ... \$ Price : num 6.95 6.75 6.75 2.1 1.25 1.65 1.25 5.95 2.55 3.75 ... \$ Customer ID: num 13085 13085 13085 13085 13085 ... \$ Country : chr "United kingdom" "United kingdom" "United kingdom" "United kingdom" ...</pre>

Hasil dari pencarian distinct value dan penentuan karaktersitik setiap atribut terdapat dalam tabel di bawah ini.

Atribut	Karakteristik
Invoice	Character
StockCode	Character
Description	Character
Quantity	Numeric
InvoiceDate	Date
Price	Numeric
Customer ID	Numeric
Country	Character

- ➔ Pada tabel di atas didapatkan 8 atribut dimana karakteristik digolongkan dalam 3 jenis yaitu :
- **Character:** Karakteristik data yang hanya menyimpan 1 digit karakter.
 - **Date:** Karakteristik data yang terbentuk dari beberapa tipe data sehingga biasa disebut dengan tipe data komposit (dapat menampung banyak nilai).
 - **Numeric:** Karakteristik data yang menyimpan nilai dari suatu atribut berupa angka (Real atau Integer)

DISTINCT VALUE

Selain itu, terdapat pencarian distinct value dimana hal ini digunakan untuk mencari nilai yang unik (tidak terdapat duplikat dalam atribut).



Syntax
<pre>#Cari distinct value length(unique(data_gabung_awal\$Invoice)) length(unique(data_gabung_awal\$StockCode)) length(unique(data_gabung_awal\$Description)) length(unique(data_gabung_awal\$Quantity)) length(unique(data_gabung_awal\$InvoiceDate)) length(unique(data_gabung_awal\$Price)) length(unique(data_gabung_awal\$`Customer ID`)) length(unique(data_gabung_awal\$Country))</pre>
Hasil
<pre>> #Cari distinct value > length(unique(data_gabung_awal\$Invoice)) [1] 53628 > length(unique(data_gabung_awal\$StockCode)) [1] 5304 > length(unique(data_gabung_awal\$Description)) [1] 5656 > length(unique(data_gabung_awal\$Quantity)) [1] 1057 > length(unique(data_gabung_awal\$InvoiceDate)) [1] 47635 > length(unique(data_gabung_awal\$Price)) [1] 2807 > length(unique(data_gabung_awal\$`Customer ID`)) [1] 5943 > length(unique(data_gabung_awal\$Country)) [1] 43</pre>

Hasil dari pencarian distinct value dan penentuan karakteristik setiap atribut terdapat dalam tabel di bawah ini.

Atribut	Distinct Value
Invoice	53628
StockCode	5304
Description	5656
Quantity	1057
InvoiceDate	47635
Price	2807
Customer ID	5943
Country	43

- ➔ Pencarian distinct value dari setiap atribut dengan menggunakan fungsi `length(unique(nama_data_frame$nama_kolom))`. Penggunaan fungsi ini akan menemukan atribut yang unik dimana tidak ada duplikat atribut dalam data tersebut. Fokus dari output distinct value adalah menghitung jumlah record yang ada pada setiap atribut dimana meskipun ada 2 record yang sama dalam 1 atribut namun tetap akan dihitung satu kesatuan (unik).

DIMENSI

Dalam menampilkan jumlah baris dan atribut yang terdapat dalam data frame maka dapat digunakan fungsi `dim(nama_data_frame)`. Atribut dapat diartikan sebagai kolom karena setiap kolom mewakili dari atribut yang ada dalam data frame. Berikut ini merupakan penerapan syntax dan hasilnya dalam menampilkan jumlah baris dan kolom.

Syntax
<pre>#dimensi data dim(data_gabung_awal)</pre>



Hasil
<pre>> dim(data_gabung_awal) [1] 1067371 8</pre>

- ➔ Pada output dimensi data dapat diketahui bahwa data Online Retail terdiri dari 1067371 baris dan 8 kolom.

HEAD & TAIL

Fungsi Head & Tail digunakan untuk menampilkan data. Head bertujuan untuk menampilkan data teratas dari suatu frame dimana biasanya data yang ditampilkan berjumlah 6 (n=6). Tail bertujuan untuk menampilkan data terbawah dimana konsepnya sama dengan Head namun hanya berbanding terbalik saja. Data yang ditampilkan berupa 6 baris teratas dan terbawah dari semua atribut yang ada dalam data tersebut. Dalam hal ini maka akan ditampilkan 6 baris teratas dan 6 baris terbawah dari 8 kolom yang ada pada data.

Penerapan fungsi head & tail ➔ head(nama_data_frame) sedangkan untuk tail yaitu tail(nama_data_frame). Di bawah ini penerapan syntax dan output yang dihasilkan.

Syntax
<pre>#head&tail data head(data_gabung_awal) tail(data_gabung_awal)</pre>
Hasil
<pre>> head(data_gabung_awal) # A tibble: 6 x 8 Invoice StockCode Description Quantity InvoiceDate Price `Customer ID` Country <chr> <chr> <chr> <dbl> <dtm> <dbl> <dbl> <chr> 1 489434 85048 15CM CHRISTMA~ 12 2009-12-01 07:45:00 6.95 13085 United~ 2 489434 79323P PINK CHERRY L~ 12 2009-12-01 07:45:00 6.75 13085 United~ 3 489434 79323W WHITE CHERRY ~ 12 2009-12-01 07:45:00 6.75 13085 United~ 4 489434 22041 "RECORD FRAME~ 48 2009-12-01 07:45:00 2.1 13085 United~ 5 489434 21232 STRAWBERRY CE~ 24 2009-12-01 07:45:00 1.25 13085 United~ 6 489434 22064 PINK DOUGHNUT~ 24 2009-12-01 07:45:00 1.65 13085 United~ > tail(data_gabung_awal) # A tibble: 6 x 8 Invoice StockCode Description Quantity InvoiceDate Price `Customer ID` Country <chr> <chr> <chr> <dbl> <dtm> <dbl> <dbl> <chr> 1 581587 22613 PACK OF 20 SP~ 12 2011-12-09 12:50:00 0.85 12680 France 2 581587 22899 CHILDREN'S AP~ 6 2011-12-09 12:50:00 2.1 12680 France 3 581587 23254 CHILDRENS CUT~ 4 2011-12-09 12:50:00 4.15 12680 France 4 581587 23255 CHILDRENS CUT~ 4 2011-12-09 12:50:00 4.15 12680 France 5 581587 22138 BAKING SET 9 ~ 3 2011-12-09 12:50:00 4.95 12680 France 6 581587 POST POSTAGE 1 2011-12-09 12:50:00 18 12680 France</pre>



SUMMARY

Dalam menampilkan ringkasan dari suatu data frame dapat menggunakan fungsi summary. Fungsi ini bertujuan untuk menampilkan hasil dari beberapa nilai statistik setiap atribut yang ada di data frame tersebut. Nilai tersebut berupa nilai minimum (Min), nilai quantil pertama (1st Qu.), Nilai tengah (Median), nilai rata-rata (Mean), nilai Quantil ketiga (3rd Qu.), dan nilai maksimum (Max).

Penerapan fungsi summary → summary(nama_data_frame).

Di bawah ini merupakan syntax summary dan hasil untuk data frame pada studi kasus Online Retail:

Syntax				
<pre>#summary summary(data_gabung_awal)</pre>				
Hasil				
Invoice	StockCode	Description	Quantity	
Length:1067371	Length:1067371	Length:1067371	Min. :	-80995.00
Class :character	Class :character	Class :character	1st Qu. :	1.00
Mode :character	Mode :character	Mode :character	Median :	3.00
			Mean :	9.94
			3rd Qu. :	10.00
			Max. :	80995.00
InvoiceDate	Price	Customer ID	Country	
Min. :2009-12-01 07:45:00	Min. : -53594.36	Min. :12346	Length:1067371	
1st Qu. :2010-07-09 09:46:00	1st Qu. : 1.25	1st Qu. :13975	Class :character	
Median :2010-12-07 15:28:00	Median : 2.10	Median :15255	Mode :character	
Mean :2011-01-02 21:13:55	Mean : 4.65	Mean :15325		
3rd Qu. :2011-07-22 10:23:00	3rd Qu. : 4.15	3rd Qu. :16797		
Max. :2011-12-09 12:50:00	Max. : 38970.00	Max. :18287		
		NA's :243007		

DESCRIBE

Fungsi describe hampir sama dengan summary dimana berguna untuk menampilkan ringkasan dari suatu data frame. Perbedaan dari keduanya yaitu output yang dihasilkan lebih lengkap pada penerapan fungsi describe apalagi untuk data numerik.

Terdapat tampilan informasi missing yang berarti ada tidaknya suatu nilai yang tidak terbaca atau tidak mempunyai nilai (missing value). Selain itu, penggunaan fungsi ini juga dapat menampilkan distinct dimana nilai unik dari setiap atribut dalam data frame. Penerapan sama dimana hanya menuliskan nama dari data frame yang akan ditampilkan. Di bawah ini merupakan syntax describe dan hasilnya:

Syntax
<pre>#deskripsi data describe(data_gabung_awal)</pre>



Hasil

```
> describe(data_gabung_awal)
```

```
data_gabung_awal
```

```
8 variables      1067371 observations
```

```
-----
```

```
Invoice
```

```
  n missing distinct
1067371      0      53628
```

```
lowest : 489434 489435 489436 489437 489438 , highest: C581484 C581490 C581499 C581568 C581569
```

```
-----
```

```
StockCode
```

```
  n missing distinct
1067371      0      5304
```

```
lowest : 10002 10002R 10080 10109 10120 , highest: POST S SP1002 TEST001 TEST002
```

```
-----
```

```
Description
```

```
  n missing distinct
1062989 4382 5655
```

```
lowest : *Boombox Ipod Classic
```

```
*USB Office Glitter Lamp
```

```
*USB Office Mirror Ball
```

```
? ? sold as sets?
```

```
highest: ZINC T-LIGHT HOLDER STARS SMALL ZINC TOP 2 DOOR WOODEN SHELF
```

```
ZINC WILLIE WINKIE CANDLE STICK
```

```
ZINC WIRE KITCHEN ORGANISER ZINC WIRE SWEETHEART LETTER TRAY
```

```
-----
```

```
Quantity
```

```
  n missing distinct
1067371      0      1057
.75 .90 .95
10 24 30
```

```
Info
```

```
Mean
```

```
Gmd
```

```
.05
```

```
.10
```

```
.25
```

```
.50
```

```
lowest : -80995 -74215 -9600 -9360 -9200, highest: 12744 12960 19152 74215 80995
```

```
value      -80000 -74000 -10000 -8000 -6000 -4000 -2000 0 2000 4000
Frequency    1 1 9 4 15 22 111 1066824 290 51
Proportion 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.999 0.000 0.000
```

```
value      6000 8000 10000 12000 20000 74000 80000
Frequency    21 5 9 5 1 1 1
Proportion 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

```
For the frequency table, variable is rounded to the nearest 2000
```

```
-----
```

```
InvoiceDate
```

```
  n missing distinct Info
1067371      0      47635 1
Mean Gmd .05 .10
2011-01-02 21:13:55 21766012 2010-01-12 17:15:00 2010-03-01 13:14:00
.25 .50 .75 .90
2010-07-09 09:46:00 2010-12-07 15:28:00 2011-07-22 10:23:00 2011-11-02 15:33:00
.95
2011-11-22 10:21:00
```

```
lowest : 2009-12-01 07:45:00 2009-12-01 07:46:00 2009-12-01 09:06:00 2009-12-01 09:08:00 2009-12-01 09:24:00
highest: 2011-12-09 12:23:00 2011-12-09 12:25:00 2011-12-09 12:31:00 2011-12-09 12:49:00 2011-12-09 12:50:00
```

```
-----
```

```
Price
```

```
  n missing distinct Info Mean Gmd .05 .10 .25 .50
1067371      0      2807 0.998 4.649 6.44 0.42 0.65 1.25 2.10
.75 .90 .95
4.15 7.95 9.95
```

```
lowest : -53594.36 -44031.79 -38925.87 -11062.06 0.00
highest: 16888.02 17836.46 18910.69 25111.09 38970.00
```

```
-----
```

```
Customer ID
```

```
  n missing distinct Info Mean Gmd .05 .10 .25 .50
824364 243007 5942 1 15325 1959 12681 12971 13975 15255
.75 .90 .95
16797 17713 17911
```

```
lowest : 12346 12347 12348 12349 12350, highest: 18283 18284 18285 18286 18287
```

```
-----
```

```
Country
```

```
  n missing distinct
1067371      0      43
```

```
lowest : Australia Austria Bahrain Belgium Bermuda
```

```
highest: United Arab Emirates United Kingdom Unspecified USA West Indies
```



DISTRIBUSI KELAS

Distribusi kelas bertujuan untuk melihat distribusi maupun presentasi dari setiap kelas suatu data frame. Dalam tabel di bawah ini hanya menampilkan salah satu contoh penerapan distribusi kelas pada kolom Country.

Penggunaan Syntac :

`y <- (data_frame)$ (nama_kolom)` → Pengambilan data dengan memilih salah satu kolom

`cbind(freq=table(y), percentage=prop.table(table(y))*100)` → untuk menampilkan jumlah frekuensi dan persentasi pada data y (data yang akan dilihat distribusi kelasnya).

Pada hasil distribusi kelas di kolom Country menunjukkan banyaknya frekuensi dari transaksi yang dilakukan pada 43 negara yang melakukan pembelian di Online Retail.

Syntax	
<pre>#distribusi kelas y <- data_gabung_awal\$Country cbind(freq=table(y), percentage=prop.table(table(y))*100)</pre>	
Hasil	
<pre>> y <- data_gabung_awal\$Country > cbind(freq=table(y), percentage=prop.table(table(y))*100)</pre>	
Australia	1913 1.792254e-01
Austria	938 8.787947e-02
Bahrain	126 1.180471e-02
Belgium	3123 2.925881e-01
Bermuda	34 3.185397e-03
Brazil	94 8.806685e-03
Canada	228 2.136090e-02
Channel Islands	1664 1.558971e-01
Cyprus	1176 1.101772e-01
Czech Republic	30 2.810644e-03
Denmark	817 7.654321e-02
EIRE	17866 1.673832e+00
European Community	61 5.714976e-03
Finland	1049 9.827886e-02
France	14330 1.342551e+00
Germany	17624 1.651160e+00
Greece	663 6.211523e-02
Hong Kong	364 3.410248e-02
Iceland	253 2.370310e-02
Israel	371 3.475830e-02
Italy	1534 1.437176e-01
Japan	582 5.452650e-02
Korea	63 5.902353e-03
Lebanon	58 5.433912e-03
Lithuania	189 1.770706e-02
Malta	299 2.801275e-02
Netherlands	5140 4.815570e-01
Nigeria	32 2.998020e-03
Norway	1455 1.363162e-01
Poland	535 5.012315e-02
Portugal	2620 2.454629e-01
RSA	169 1.583330e-02
Saudi Arabia	10 9.368814e-04
Singapore	346 3.241610e-02
Spain	3811 3.570455e-01
Sweden	1364 1.277906e-01
Switzerland	3189 2.987715e-01
Thailand	76 7.120298e-03
United Arab Emirates	500 4.684407e-02
United Kingdom	981330 9.193898e+01
Unspecified	756 7.082823e-02
USA	535 5.012315e-02
West Indies	54 5.059159e-03

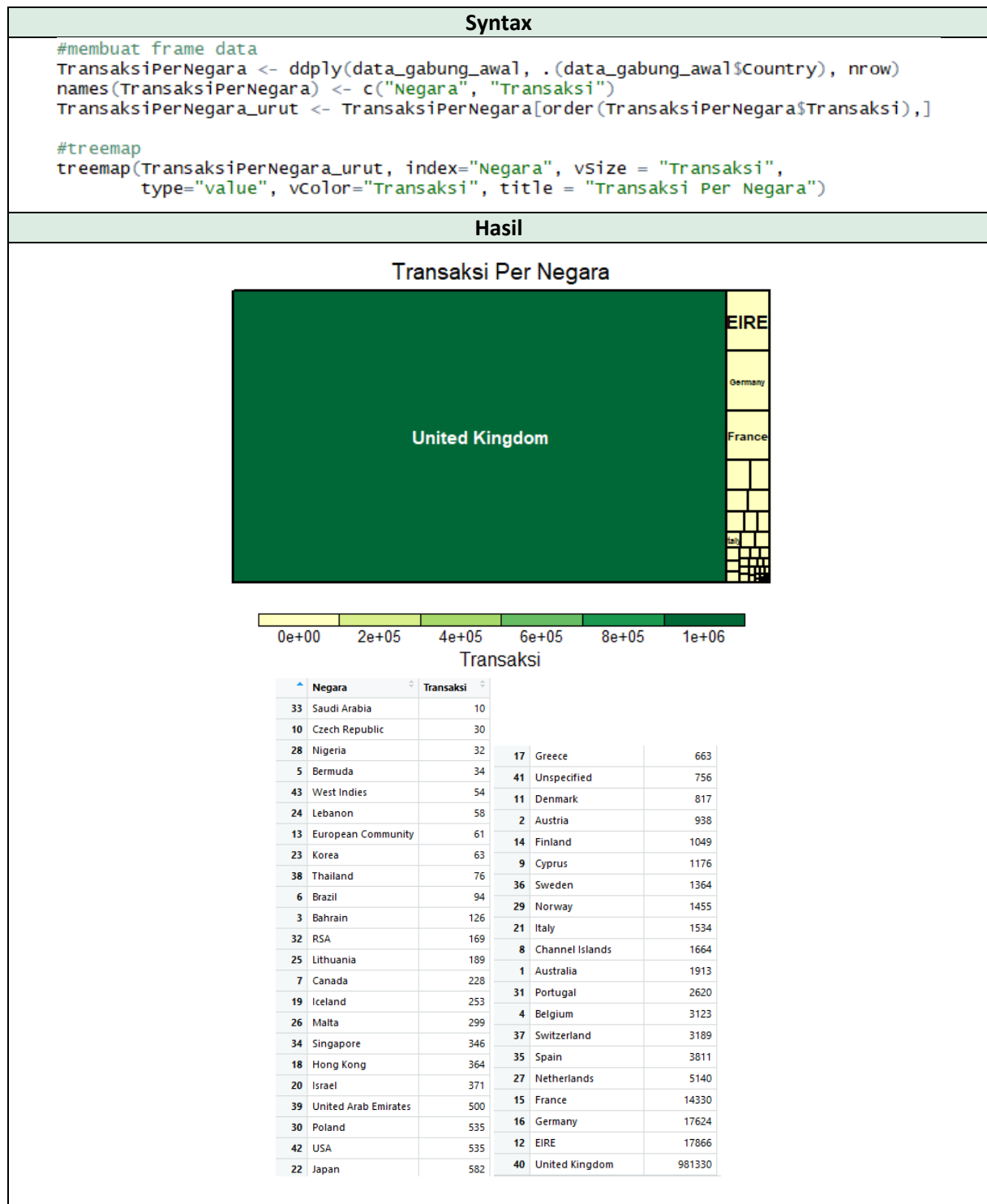


VISUALISASI DATA

TRANSAKSI PER NEGARA

Karakteristik data yang pertama dianalisis adalah jumlah transaksi per negara. Dalam merepresentasikan data, treemap digunakan untuk melihat data secara keseluruhan agar mengetahui proporsi transaksi masing-masing negara, kemudian bar plot digunakan untuk melihat 5 negara dengan transaksi tertinggi dan terendah.

- **Treemap Keseluruhan**



Dari hasil treemap yang ditampilkan, dapat diketahui bahwa transaksi pada negara Inggris memiliki porsi yang paling besar dibandingkan dengan negara-negara lainnya.

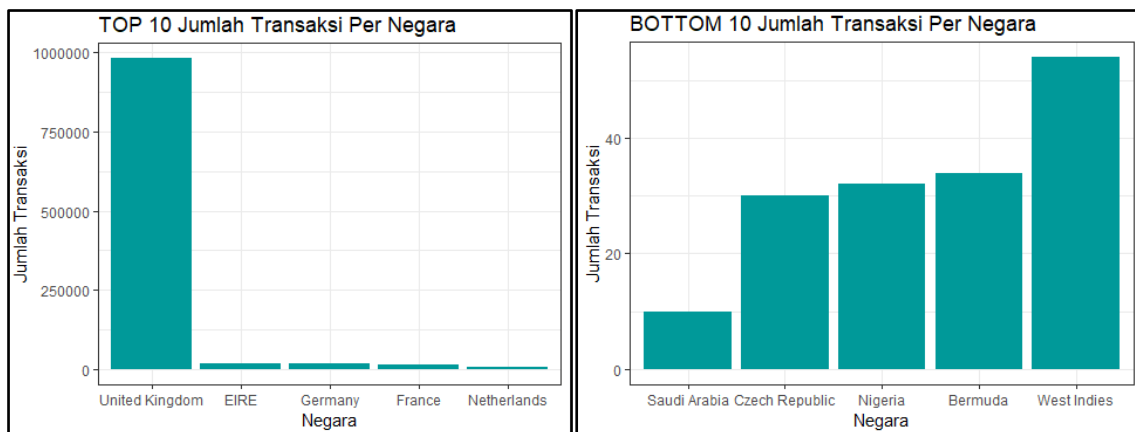
- **TOP 10**

Syntax

```
#TOP 10
TransaksiPerNegara_TOP10 <- TransaksiPerNegara_urut[39:43,]
ggplot(TransaksiPerNegara_TOP10, aes(x=reorder(Negara,-Transaksi), y=Transaksi)) +
  geom_bar(stat="identity", fill="#009999") + theme_bw() +
  xlab("Negara") +
  ylab("Jumlah Transaksi") +
  ggtitle("TOP 10 Jumlah Transaksi Per Negara")

#BOTT 10
TransaksiPerNegara_BOT10 <- TransaksiPerNegara_urut[1:5,]
ggplot(TransaksiPerNegara_BOT10, aes(x=reorder(Negara,Transaksi), y=Transaksi)) +
  geom_bar(stat="identity", fill="#009999") + theme_bw() +
  xlab("Negara") +
  ylab("Jumlah Transaksi") +
  ggtitle("BOTTOM 10 Jumlah Transaksi Per Negara")
```

Hasil



Bar plot diatas menggambarkan tentang 5 negara dengan transaksi tertinggi dan terendah. Lima negara dengan transaksi tertinggi adalah United Kingdom, EIRE, Jerman, Prancis dan Belanda. Sedangkan negara dengan transaksi terendah adalah Saudi Arabia, Republik Ceko, Nigeria, Bermuda dan West Indies. Dari informasi tersebut dapat diketahui bahwa penjualan barang cenderung terjadi pada negara-negara di Benua Eropa.



JUMLAH PELANGGAN UNIK DI TIAP NEGARA

Karakteristik selanjutnya yang perlu diketahui adalah jumlah pelanggan (unik) di tiap negara. Maksud dari “unik” disini adalah seorang pelanggan tetap dihitung sebagai satu individu yang sama meskipun melakukan transaksi beberapa kali. Informasi ini perlu diketahui perusahaan untuk mengetahui seberapa luas penyebaran pelanggan yang dimilikinya. Pie Chart dipilih untuk merepresentasikan informasi tersebut untuk mempermudah informasi proporsi persebaran jumlah pelanggan.

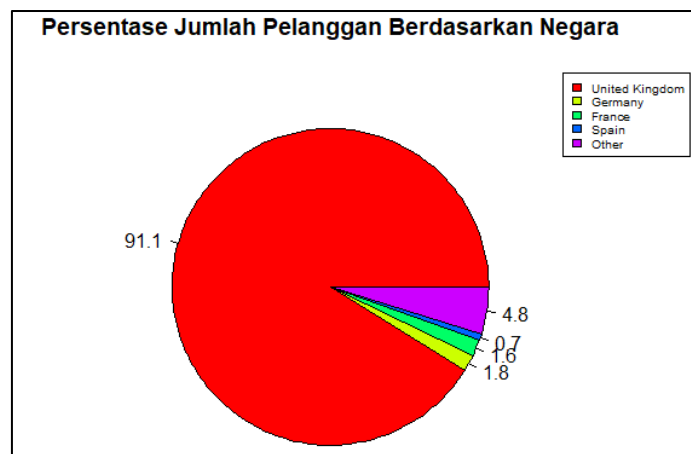
Syntax

```
#2. Jumlah pelanggan unik di tiap negara
#buat data
customerbyCountry <- dplyr::ddply(data_gabung_awa1, .(Country, `Customer ID`), nrow)
names(customerbyCountry) <- c("Country", "CustomerID", "Frequency")
customerbyCountry <- dplyr::ddply(customerbyCountry, .(customerbyCountry$Country), nrow)
names(customerbyCountry) <- c("Country", "Number of Unique Customer")
customerbyCountry <- customerbyCountry[order(-customerbyCountry$`Number of Unique Customer`),]
customerbyCountryclean <- head(customerbyCountry, 4)
customerbyCountryclean[nrow(customerbyCountryclean) + 1,] = c("other", sum(customerbyCountry$`Number of Unique Customer`[6:43]))
customerbyCountryclean$`Number of Unique Customer` <- as.numeric(customerbyCountryclean$`Number of Unique Customer`)
#buat pie
piepercent <- round(100 * customerbyCountryclean$`Number of Unique Customer` / sum(customerbyCountryclean$`Number of Unique Customer`), 1)
pie(customerbyCountryclean$`Number of Unique Customer`, main = "Persentase Jumlah Pelanggan Berdasarkan Negara",
    labels = piepercent,
    col = rainbow(length(customerbyCountryclean$`Number of Unique Customer`)))
legend("topright", customerbyCountryclean$Country, cex = 0.6,
    fill = rainbow(length(customerbyCountryclean$`Number of Unique Customer`)))
```

Hasil

Country	Number of Unique Customer	Country	Number of Unique Customer
Bermuda	1	Poland	6
Czech Republic	1	Unspecified	8
European Community	1	USA	9
Hong Kong	1	Japan	10
Iceland	1	Cyprus	11
Lithuania	1	Denmark	12
Saudi Arabia	1	Austria	13
Singapore	1	Norway	13
Thailand	1	Channel Islands	14
West Indies	1	Australia	15
Brazil	2	Finland	15
Korea	2	Italy	17
Lebanon	2	Sweden	20
Malta	2	Netherlands	23
Nigeria	2	Switzerland	23
Bahrain	3	Portugal	25
RSA	3	Belgium	29
Canada	5	Spain	41
Greece	5	France	96
Israel	5	Germany	107
United Arab Emirates	5	United Kingdom	5411

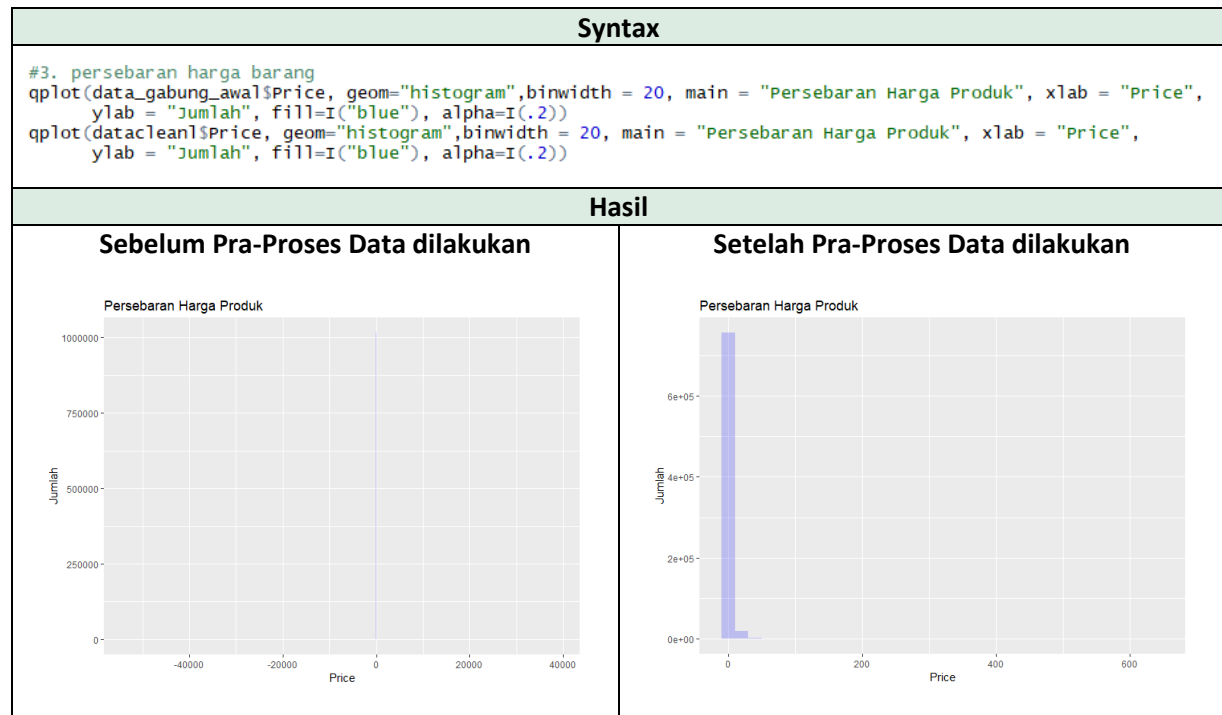
Persentase Jumlah Pelanggan Berdasarkan Negara



Dari hasil visualisasi diatas, didapatkan informasi mengenai persentase jumlah pelanggan di tiap negara. Diketahui bahwa United Kingdom memiliki jumlah peanggan yang paling besar yaitu berjumlah 5411 (91,1%) dan diikuti Jerman dengan 107 (1,8%) pelanggan, Perancis dengan 96(1,6%) pelanggan dan Spanyol sebanyak 41 (0,7%) pelanggan dan 4,8% sisanya diambil dari negara-negara lainnya.

PERSEBARAN HARGA BARANG YANG DIJUAL

Selanjutnya karakter dataset yang dianalisis adalah pada persebaran harga barang yang dijual pada Online Retail tersebut. Histogram digunakan untuk merepresentasikan persebaran tersebut.

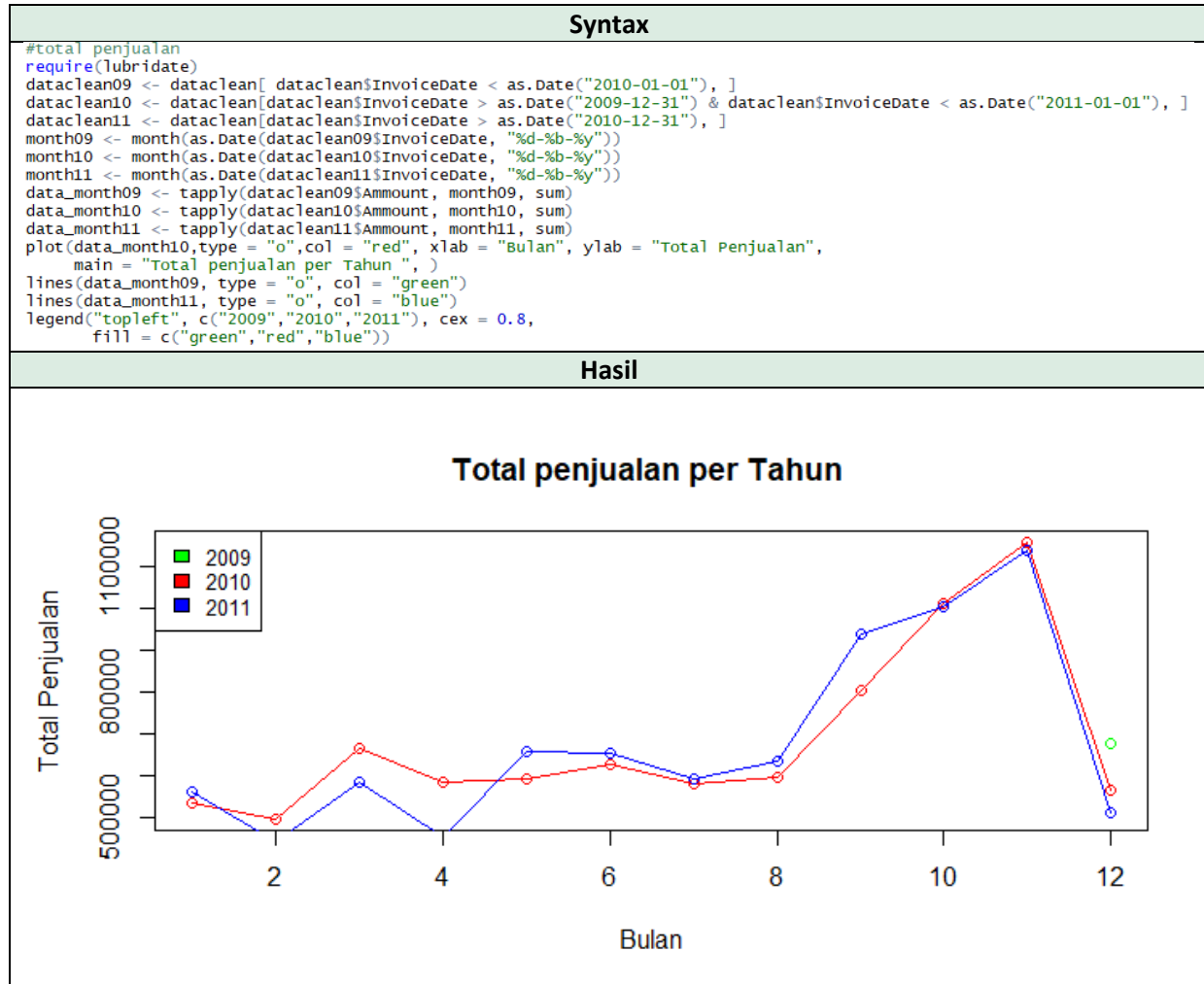


Dari hasil visualisasi diatas, didapatkan informasi mengenai persebaran harga barang yang dijual, dapat dilihat sebelum dilakukan pra-proses data, barang yang dijual masih ada yang bernilai negative dan ada yang terlampau tinggi hingga > 400.000 Sterling. Garis tipis sekitar angka 0 menunjukkan bahwa banyak barang yang berharga kisaran 0-100. Setelah data dikurasi pada langkah pra-proses data, didapatkan informasi yang lebih jelas bahwa harga barang yang dijual antara 0 Sterling (gratis) hingga kurang lebih 600 Sterling. Namun, kebanyakan barang (± 750.000 barang) yang dijual berharga dari 0 hingga 100 Sterling.



TOTAL PENJUALAN PER BULAN PADA TAHUN 2009-2011

Analisis total penjualan per bulan dapat dilihat untuk mengetahui tren penjualan perusahaan. Line chart dipilih untuk merepresentasikan informasi tersebut agar data time-series dapat ditampilkan dengan jelas sesuai dengan urutannya.



Dari hasil visualisasi tersebut, dapat diketahui bahwa ada tren kenaikan dan penurunan yang mirip pada tiap tahunnya. Penjualan akan meningkat pada bulan September, Oktober dan November. Setelahnya, terjadi penurunan pada bulan Desember.



KUANTITAS PEMBELIAN TIAP PELANGGAN

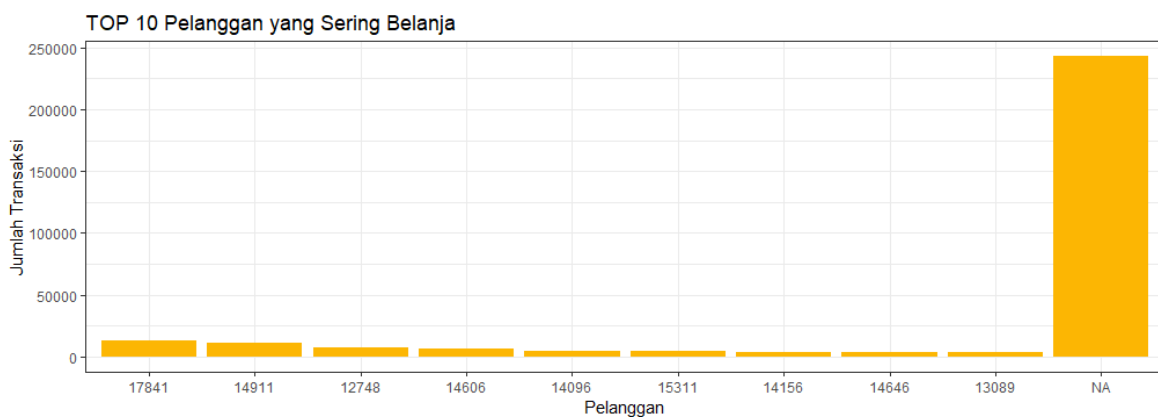
Kuantitas pembelian tiap pelanggan perlu diketahui untuk melihat seberapa sering pelanggan melakukan transaksi pada online retail tersebut. Bar plot dipilih untuk merepresentasikan informasi ini agar urutan kuantitas dapat terlihat dengan baik.

Syntax

```
#5. amount pelanggan
TransaksiPerPelanggan <- ddply(data_gabung_awal, .(data_gabung_awal$`Customer ID`), nrow)
names(TransaksiPerPelanggan) <- c("ID Pelanggan", "Total Transaksi")
TransaksiPerPelanggan_urut <- TransaksiPerPelanggan[order(-TransaksiPerPelanggan$`Total Transaksi`),]
TransaksiPerPelanggan_TOP10 <- TransaksiPerPelanggan_urut[1:10,]
TransaksiPerPelanggan_TOP10$`ID Pelanggan` <- as.character(TransaksiPerPelanggan_TOP10$`ID Pelanggan`)

ggplot(TransaksiPerPelanggan_TOP10, aes(x=reorder(`ID Pelanggan`, -`Total Transaksi`), y=`Total Transaksi`)) +
  geom_bar(stat="identity", fill="#fcb603") + theme_bw() +
  xlab("Pelanggan") +
  ylab("Jumlah Transaksi") +
  ggtitle("TOP 10 Pelanggan yang Sering Belanja")
```

Hasil



ID Pelanggan	Total Transaksi
NA	243007
17841	13097
14911	11613
12748	7307
14606	6709
14096	5128
15311	4717
14156	4130
14646	3890
13089	3438

Dari visualisasi yang dibuat, total transaksi paling banyak adalah dengan ID Pelanggan NA dimana berarti Customer ID pada dataset ada yang bernilai NULL. Hal ini merupakan informasi awal untuk perlakuan *missing values* pada tahap berikutnya. Selain itu dapat diketahui bahwa, customer ber-ID 17841 telah melakukan transaksi paling banyak dari pelanggan lainnya yaitu sebanyak 13097 kali. Sedangkan, untuk nilai minimal transaksi yang dilakukan adalah berjumlah 1 per pelanggan.



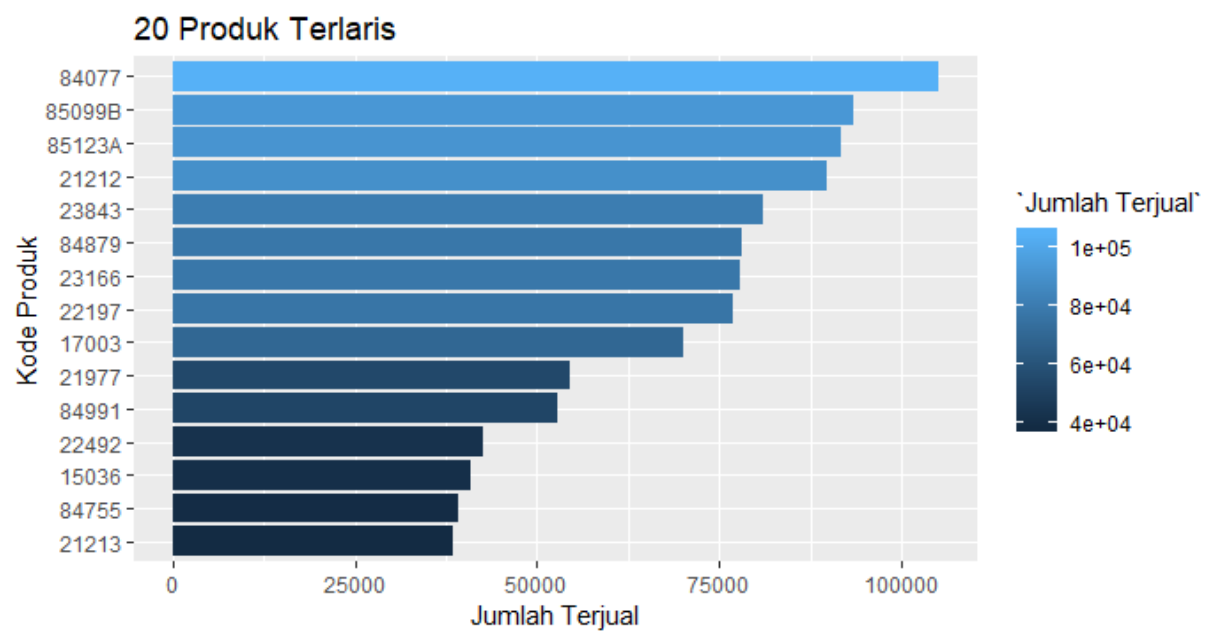
JUMLAH BARANG YANG TERJUAL

Jumlah barang yang terjual dapat ditunjukkan pada barplot dibawah.

Syntax

```
ProductbyQuantity <- ddply(data_gabung_awal,.(StockCode),summarise,'Jumlah Terjual'= sum(Quantity))
ProductbyQuantity <- ProductbyQuantity[order(-ProductbyQuantity$`Jumlah Terjual`),]
ProductbyQuantityTOP20 <- ProductbyQuantity[1:20,]
ggplot(ProductbyQuantityTOP15, aes(x=reorder(StockCode,'Jumlah Terjual'), y='Jumlah Terjual',fill='Jumlah Terjual')) +
  geom_bar(stat="identity") + coord_flip() +
  xlab("Kode Produk") +
  ylab("Jumlah Terjual") +
  ggtitle("20 Produk Terlaris")
```

Hasil



Dari hasil visualisasi yang ditampilkan, produk yang paling sering dibeli adalah berkode 84077 yakni telah terjual sebanyak lebih dari 100.000 barang. Kemudian diikuti barang berkode 85099B, 85123A dan seterusnya seperti yang ada pada barplot diatas.



BOXPLOT HARGA & KUANTITAS BARANG

Boxplot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang bisa menggambarkan bentuk distribusi data (skewness), ukuran tendensi sentral dan ukuran penyebaran (keragaman) data pengamatan.

Terdapat 5 ukuran statistik yang bisa kita baca dari boxplot, yaitu:

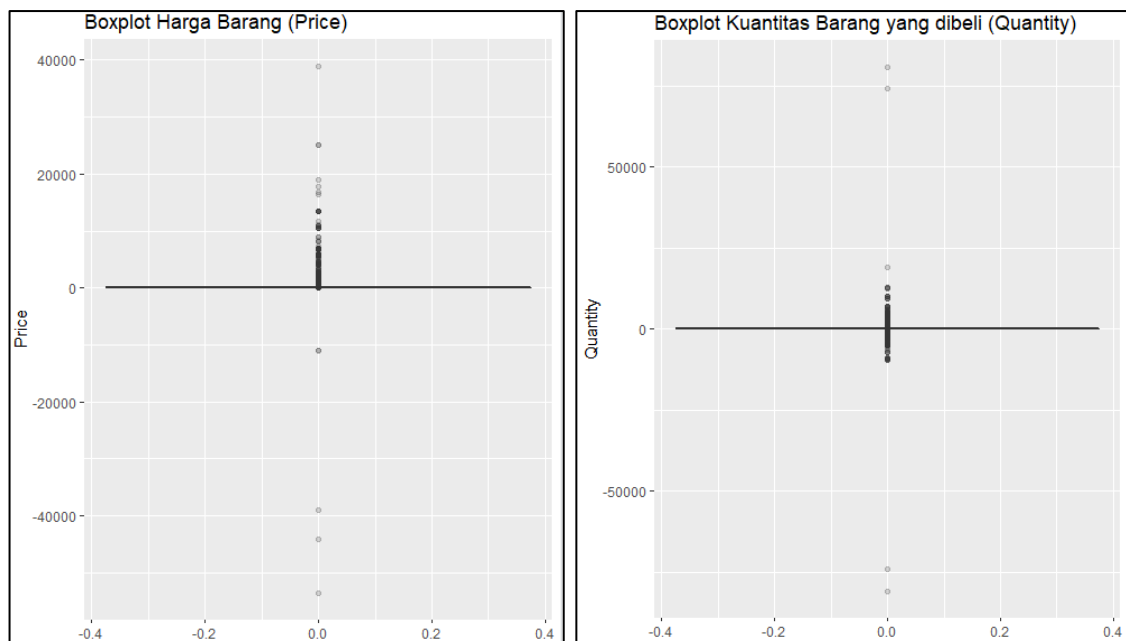
- nilai minimum: nilai observasi terkecil
- Q1: kuartil terendah atau kuartil pertama
- Q2: median atau nilai pertengahan
- Q3: kuartil tertinggi atau kuartil ketiga
- nilai maksimum: nilai observasi terbesar.

Selain itu, boxplot juga dapat menunjukkan ada tidaknya nilai outlier dan nilai ekstrim dari data pengamatan.

Syntax

```
#4. Boxplot
ggplot(data_gabung_awal, aes(y=Price, fill=Price)) +
  geom_boxplot(varwidth = TRUE, alpha=0.2) +
  theme(legend.position="none") +
  ggtitle("Boxplot Harga Barang (Price)")
ggplot(data_gabung_awal, aes(y=Quantity, fill=Price)) +
  geom_boxplot(varwidth = TRUE, alpha=0.2) +
  theme(legend.position="none") +
  ggtitle("Boxplot Kuantitas Barang yang dibeli (Quantity)")
```

Hasil



Dapat diketahui pada kedua boxplot diatas bahwa persebaran nilai dari harga (UnitPrice) dan Kuantitas Barang (Quantity) masih belum seimbang dan masih memiliki outlier. Informasi tersebut dapat dijadikan acuan untuk melakukan tahap pra-proses data selanjutnya.



BAB II : PRA-PROSES DATA

MENGGABUNGKAN DATA

Langkah awal adalah menggabungkan data pada tahun 2009-2010 dengan data pada tahun 2010-2011 yang berada pada *sheet excel* yang berbeda. Penggabungan data secara vertikal (penambahan baris) dapat dilakukan dengan menggunakan fungsi `rbind()`.

Syntax (Penggabungan Data)									
<pre>#Menggabungkan Dataset data_gabung_awal <- rbind(online_retail_2009_2010, online_retail_2010_2011)</pre>									
Hasil (Penggabungan Data)									
<table><thead><tr><th colspan="2">Data</th></tr></thead><tbody><tr><td>data_gabung_awal</td><td>1067371 obs. of 8 variables</td></tr><tr><td>online_retail_20...</td><td>525461 obs. of 8 variables</td></tr><tr><td>online_retail_20...</td><td>541910 obs. of 8 variables</td></tr></tbody></table>		Data		data_gabung_awal	1067371 obs. of 8 variables	online_retail_20...	525461 obs. of 8 variables	online_retail_20...	541910 obs. of 8 variables
Data									
data_gabung_awal	1067371 obs. of 8 variables								
online_retail_20...	525461 obs. of 8 variables								
online_retail_20...	541910 obs. of 8 variables								

Data awal yang terbentuk adalah berjumlah 1067371 baris dengan 8 variabel.

DATA DUPLIKAT

Langkah berikutnya dalam melakukan pra-proses data, harus ada pengecekan dan penindakanlanjutan untuk data-data yang sifatnya duplikat atau redundan. Untuk melakukan hal tersebut, diperlukan library `dplyr` dengan fungsi `distinct()`.

Syntax (Pengecekan Data Dupikat)					
<pre>#P2. Redundan data_noduplikat <- data_gabung_awal %>% distinct()</pre>					
Hasil (Pengecekan Data Dupikat)					
Sebelum	<table><thead><tr><th colspan="2">Data</th></tr></thead><tbody><tr><td>data_gabung_awal</td><td>1067371 obs. of 8 variables</td></tr></tbody></table>	Data		data_gabung_awal	1067371 obs. of 8 variables
Data					
data_gabung_awal	1067371 obs. of 8 variables				
Sesudah	<table><thead><tr><th colspan="2">Data</th></tr></thead><tbody><tr><td>data_noduplikat</td><td>1033036 obs. of 8 variables</td></tr></tbody></table>	Data		data_noduplikat	1033036 obs. of 8 variables
Data					
data_noduplikat	1033036 obs. of 8 variables				

Dari hasil analisis data duplikat pada dataset yang tersedia yaitu pada data online retail, ditemukan adanya data yang sifatnya duplikat. Dapat dilihat dari pengurangan yang terjadi dari data yang memiliki baris berjumlah 1067371 menjadi 1033036.



MISSING VALUE

Hal kedua yang perlu diberi perhatian adalah data-data yang berisi nilai N/A atau NULL. Pengecekan tersebut dapat dilakukan dengan fungsi `is.na()`.

Syntax (Missing Value)
<pre>#P3. Missing value #-apakah ada? any(is.na(data_noduplikat)) #-dimana aja? sapply(data_noduplikat, function(x) any(is.na(x))) #-berapa yang N/A? sapply(data_noduplikat, function(x) sum(is.na(x)))</pre>
Hasil (Missing Value)
<pre>> #-apakah ada? > any(is.na(data_noduplikat)) [1] TRUE > #-dimana aja? > sapply(data_noduplikat, function(x) any(is.na(x))) Invoice StockCode Description Quantity InvoiceDate Price Customer ID Country FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE > #-berapa yang N/A? > sapply(data_noduplikat, function(x) sum(is.na(x))) Invoice StockCode Description Quantity InvoiceDate Price Customer ID Country 0 0 4275 0 0 0 235151 0</pre>
<p>Dari hasil diatas yang dapat dilihat bahwa dari ada 2 kolom yang memiliki nilai null, yaitu Customer ID dan Descroption. Kedua atribut ini akan ditindak lanjuti di proses selanjutnya.</p>

MENGHILANGKAN DATA YANG TIDAK DIPERLUKAN

Langkah selanjutnya adalah perlu menghilangkan data-data yang tidak diperlukan untuk proses clustering.

- Menghilangkan data *Cancelled*

Data Cancelled dihilangkan dengan mendeteksi huruf "C" pada nomor invoice.

Syntax
<pre>#remove leading and trailing function data_noduplikat\$Invoice = as.character(data_noduplikat\$Invoice) trim = function(x) gsub("^\\s+ \\s+\$", "", x) data_noduplikat\$Invoice = trim(data_noduplikat\$Invoice) data_noduplikat\$Description = trim(as.character(data_noduplikat\$Description)) #menghilangkan c is_c = function(x) startswith(x,"c") data_tanpacancel = data_noduplikat[which(!is_c(data_noduplikat\$Invoice)),] #subsetting</pre>
Hasil
<p>Hasil dari penghilangan data <i>cancelled</i>, data menjadi hanya berjumlah 1013932 baris.</p>
<pre>data_tanpacancel 1013932 obs. of 8 variables</pre>



- **Menghilangkan data NULL (Missing Values)**

Data null yang sebelumnya sudah dideteksi kemudian dilakukan penghilangan agar dapat dilakukan tindakan clustering pada data.

Syntax
<pre>#hilangin null data_tanpaNULLDesc = subset(data_tanpacancel,!is.na(data_tanpacancel\$Description)) #subsetting data_tanpaNULLCust = subset(data_tanpacancel,!is.na(data_tanpacancel\$`Customer ID`)) #subsetting</pre>
Hasil
<p>Data hasil penghilangan nilai null memiliki baris berjumlah 779495 baris.</p> <pre>data_tanpaNULLCust 779495 obs. of 8 variables data_tanpaNULLDesc 1009657 obs. of 8 variables</pre>

- **Menghilangkan data yang bernilai negatif**

Setelah dilakukan penghilangan beberapa data sebelumnya, maka data negatif perlu juga dihilangkan mengingat sebelumnya pada visualisasi yang terbentuk terlihat jelas adanya data barang yang memiliki harga negatif.

Syntax
<pre>#hapus yang negatif data_tanpanegatif <- data_tanpaNULLCust[which(data_tanpaNULLCust\$Quantity>=0 & data_tanpaNULLCust\$Price >=0),]</pre>
Hasil
<p>Hasil dari penghilangan data ternyata tidak menunjukkan perubahan baris, hal ini berarti data-data negatif sebelumnya sudah tercakup dalam penghilangan data lainnya.</p> <pre>data_tanpanegatif 779495 obs. of 8 variables</pre>

- **Menghilangkan data “Buzzword”**

Langkah selanjutnya adalah menghilangkan data yang mengandung “Buzzword”. Buzzword yang dimaksud adalah seperti AWAY, FEE, CRASHED, MAILOUT, dan sebagainya.

Syntax
<pre>#menghilangkan buzzword is_buzzword = function(x) { str_detect(toupper(x),"AWAY") str_detect(toupper(x),"CHARGES") str_detect(toupper(x),"FEE") str_detect(toupper(x),"FAULT") str_detect(toupper(x),"SALES") str_detect(toupper(x),"ADJUST") str_detect(toupper(x),"COUNTED") str_detect(toupper(x),"INCORRECT") str_detect(toupper(x),"WRONG") str_detect(toupper(x),"LOST") str_detect(toupper(x),"CRUSHED") str_detect(toupper(x),"DAMAGE") str_detect(toupper(x),"FOUND") str_detect(toupper(x),"THROWN") str_detect(toupper(x),"SMASHED") str_detect(toupper(x),"\\?") str_detect(toupper(x),"BROKEN") str_detect(toupper(x),"BARCODE") str_detect(toupper(x),"RETURNED") str_detect(toupper(x),"MAILOUT") </pre>



```

str_detect(toupper(x), "DELIVERY") | str_detect(toupper(x), "MIX UP") |
str_detect(toupper(x), "MOULDY") | str_detect(x, "Bank") |
str_detect(toupper(x), "PUT ASIDE") | str_detect(toupper(x), "ERROR") |
str_detect(toupper(x), "DESTROYED") | str_detect(toupper(x), "RUSTY") |
str_detect(toupper(x), "MANUAL") | str_detect(toupper(x), "AMAZON") |
str_detect(toupper(x), "POSTAGE") | str_detect(toupper(x), "PADS")
}
data_tanpabuzzword = data_tanpanegatif[which(!is Buzzword(as.character(data_tanpanegatif$Description))),]

```

Hasil

Dari penghilangan buzzword yang dilakukan, data yang dihasilkan memiliki baris berjumlah 776904.

 data_tanpabuzzword | 776904 obs. of 8 variables

FORMAT DATA WAKTU

Setelah penghilangan data yang tidak perlu, selanjutnya dilakukan format untuk data tanggal invoice. Format waktu tersebut dilakukan untuk mengubah data jam-tanggal menjadi data tanggal saja.

Syntax

```

#format waktu
Time = format(as.POSIXct(strptime(data_tanpabuzzword$Invoice, "%Y-%m-%d %H:%M", tz="")), format = "%H:%M:%S")
data_tanpabuzzword$InvoiceDate = as.Date(data_tanpabuzzword$InvoiceDate)
dataclean <- data_tanpabuzzword

```

Hasil

Sebelum :

InvoiceDate
2009-12-01 07:45:00
2009-12-01 07:45:00
2009-12-01 07:45:00
2009-12-01 07:45:00
2009-12-01 07:45:00

Sesudah :

InvoiceDate
2009-12-01
2009-12-01
2009-12-01
2009-12-01
2009-12-01

SAMPLING

Untuk menanggulangi memori laptop yang tidak cukup dalam pengolahan data, maka perlu adanya proses sampling. Metode sampling yang digunakan pada kali ini adalah dengan random sampling. Metode ini dapat dilakukan dengan memanggil fungsi sample().

Jika merujuk pada rumus statistika sampling menggunakan metode slovin, sebagai berikut :




$$n = \frac{N}{(1 + N \times e^2)}$$

Dengan menggunakan tingkat kepercayaan (confidence level) sebesar 95%, maka :

$$n = \frac{776904}{(1 + 776904 \times 0.05^2)} = 399.7941603$$

Dari perhitungan tersebut dapat disimpulkan bahwa jumlah sample minimal yang dibutuhkan adalah $399.7941603 \cong 400$. Namun dengan pertimbangan spesifikasi laptop yang digunakan adalah RAM 4 GB dan intel i5, maka kami menggunakan sample sebanyak 10000 data.



Syntax					
<pre>#sampling datasample <- dataclean[sample(nrow(dataclean), 10000),]</pre>					
Hasil					
<p>Data yang akan diproses selanjutnya berjumlah baris 10.000 (setelah dilakukan penyesuaian dengan spesifikasi laptop)</p>					
<table> <tr> <th colspan="2">Data</th></tr> <tr> <td> datasample</td><td>10000 obs. of 10 variables</td></tr> </table>		Data		 datasample	10000 obs. of 10 variables
Data					
 datasample	10000 obs. of 10 variables				

PENANGANAN OUTLIER PERTAMA

Outlier adalah data observasi yang muncul dengan nilai-nilai ekstrim, baik secara univariat ataupun multivariat. Yang dimaksud dengan nilai-nilai ekstrim dalam observasi adalah nilai yang jauh atau beda sama sekali dengan sebagian besar nilai lain dalam kelompoknya.

Syntax	
<pre>#outlier library(OutlierDetection) outlier <- nn(datasample, k=4)\$`Location of outlier` data_nooutlier<-datasample[-c(outlier),]</pre>	
Hasil	
<p>Berikut merupakan baris-baris yang terdeteksi sebagai outlier pada dataset dan perlu dihilangkan agar clustering yang dibentuk bagus.</p>	
<pre>> outlier [1] 3 17 21 52 73 124 140 177 191 209 234 238 288 295 302 [16] 323 328 332 347 352 356 377 430 446 449 468 504 525 539 549 [31] 563 565 635 664 665 672 680 721 750 752 789 816 817 828 836 [46] 854 889 940 948 960 971 975 984 998 1027 1041 1051 1057 1092 1143 [61] 1151 1160 1169 1186 1251 1258 1261 1273 1280 1290 1338 1380 1418 1506 1522 [76] 1539 1600 1604 1635 1644 1657 1692 1718 1760 1767 1771 1779 1786 1828 1864 [91] 1941 1943 1948 1999 2016 2022 2088 2113 2114 2118 2148 2208 2218 2221 2231 [106] 2233 2243 2247 2250 2273 2286 2386 2407 2412 2422 2443 2447 2464 2481 2506 [121] 2515 2518 2527 2538 2547 2564 2573 2579 2597 2609 2636 2717 2773 2867 2907 [136] 2944 2971 2993 3015 3081 3086 3088 3101 3127 3149 3155 3159 3193 3225 3248 [151] 3259 3331 3346 3366 3461 3465 3483 3489 3507 3510 3511 3516 3520 3576 3580 [166] 3608 3612 3650 3665 3669 3687 3688 3691 3710 3730 3738 3769 3771 3772 3779 [181] 3805 3847 3859 3931 3932 3934 3952 3969 3975 3977 3978 4037 4052 4055 4132 [196] 4143 4223 4224 4240 4254 4260 4265 4297 4299 4309 4315 4349 4356 4377 4438 [211] 4463 4513 4515 4565 4590 4606 4664 4731 4732 4750 4769 4770 4786 4803 4810 [226] 4829 4832 4882 4927 4931 4982 4993 4998 5048 5055 5078 5099 5103 5159 5203 [241] 5204 5267 5316 5318 5352 5422 5455 5463 5471 5495 5500 5503 5514 5530 5544 [256] 5550 5552 5588 5590 5611 5617 5625 5629 5633 5641 5663 5838 5858 5891 5932 [271] 5938 5952 5963 5964 5966 5989 5992 6019 6021 6038 6054 6067 6071 6075 6081 [286] 6096 6131 6132 6162 6190 6216 6224 6244 6256 6284 6314 6371 6382 6384 6416 [301] 6418 6427 6433 6440 6446 6454 6472 6494 6502 6532 6570 6577 6589 6596 6644 [316] 6682 6733 6747 6753 6761 6765 6817 6818 6826 6833 6836 6848 6870 6881 6890 [331] 6919 6943 6956 6960 7010 7021 7031 7058 7067 7092 7137 7143 7155 7173 7236 [346] 7253 7290 7343 7389 7466 7526 7549 7554 7592 7603 7665 7668 7710 7720 7732 [361] 7736 7739 7743 7752 7763 7821 7827 7828 7854 7903 7906 7932 8000 8045 8050 [376] 8053 8086 8109 8135 8137 8153 8162 8164 8166 8179 8198 8221 8225 8234 8292 [391] 8299 8339 8356 8359 8415 8418 8472 8475 8479 8493 8519 8533 8609 8616 8653 [406] 8743 8764 8769 8799 8846 8852 8855 8856 8862 8892 8913 8941 8946 8965 8978 [421] 9054 9059 9064 9089 9101 9122 9136 9184 9225 9243 9266 9277 9278 9297 9300 [436] 9328 9334 9358 9360 9410 9413 9414 9428 9438 9441 9456 9463 9505 9553 9708 [451] 9715 9738 9805 9821 9849 9865 9866 9875 9895 9901 9908 9909 9915 9959 9998 [466] 9999</pre>	



MEMBUAT BENTUK DATA RFM

Untuk melakukan analisis RFM, data perlu dibentuk menjadi 3 kategori yaitu Recency, Frequency dan Monetary seperti berikut.

Pada tahap ini sebelum dilakukan proses kategori sesuai analisis RFM maka perlu dibuat data frame customers yang nantinya akan berisi Customer ID, recency, frequency, dan monetary. Berikut ini merupakan penjelasan dari setiap pembuatan kolom RFM:

- Recency: Pembuatan kolom recency yang berarti penentuan tanggal 10 Desember 2011 sebagai tanggal acuan dimana nantinya akan dikurangi dengan tanggal transaksi terakhir dilakukan (invoice date).
- Frequency: Pembuatan kolom frequency yang berarti mencatat jumlah transaksi yang dilakukan berdasarkan customer id. Jadi setiap transaksi dalam melakukan setiap transaksi akan dijumlah sehingga dapat terlihat berdasarkan customer id yang memiliki jumlah transaksi.
- Monetary: Pembuatan kolom monetary yang berarti mencatat jumlah uang yang dikeluarkan oleh setiap customer berdasarkan customer id. Ammount ditentukan dari perkalian antara price dengan quantity yang didapatkan dari setiap transaksi oleh customer. Sedangkan monetary dihasilkan dari penjumlahan dari Ammount.

Syntax

```
#####
# Create customer-level dataset #
#####

customers <- as.data.frame(unique(data_nooutlier$`Customer ID`))
names(customers) <- "CustomerID"
customers$CustomerID <- customers[order(customers$CustomerID),]

#R - RECENCY
data_nooutlier$recency <- as.Date("2011-12-10") - as.Date(data_nooutlier$InvoiceDate)
recency <- aggregate(recency ~ `Customer ID`,
                     data=data_nooutlier, FUN=min, na.rm=TRUE)
customers <- merge(customers, recency,
                   by.x="CustomerID",by.y="Customer ID", all=TRUE, sort=TRUE)
remove(recency)
customers$recency <- as.numeric(customers$recency)

#F - FREQUENCY
frequency <- ddply(data_nooutlier,.(data_nooutlier$`Customer ID`), nrow)
names(frequency) <- c("CustomerID", "frequency")
frequency <- frequency[order(-frequency$CustomerID),]
customers <- merge(customers, frequency,
                   by.x="CustomerID",by.y="CustomerID", all=TRUE, sort=TRUE)
remove(frequency)

#M - Monetary
dataclean['Ammount'] = dataclean['Price']*dataclean['Quantity']
monetary <- ddply(data_nooutlier,.(`Customer ID`),summarise,monetary= sum(Ammount))
monetary <- monetary[order(-monetary$`Customer ID`),]
customers <- merge(customers, monetary,
                   by.x="CustomerID",by.y="Customer ID", all=TRUE, sort=TRUE)
remove(monetary)
```

Hasil

	CustomerID	recency	frequency	monetary
1	12347	40	1	17.00
2	12349	19	3	65.10
3	12350	311	1	17.40
4	12352	73	1	19.80



PENANGANAN OUTLIER KEDUA

Pada saat data sudah dimasukkan dalam kategori RFM ternyata masih ditemukan data yang memiliki outlier. Oleh karena itu, perlu menghilangkan outlier sehingga proses clustering dapat menghasilkan cluster yang baik dimana tidak ada data yang tidak memiliki cluster.

Syntax	
<pre>#outlier library(OutlierDetection) coutlier <- nn(customers, k=3)\$`Location of outlier` customers<-customers[-c(coutlier),]</pre>	
Hasil	
customers	2909 obs. of 4 variables

MEMBUAT DATA CLUSTER DAN SCALING

Pembuatan data cluster merupakan langkah membuat data yang tanpa berisi kolom Customer ID sehingga data customercluster hanya berisi kolom recency, frequency, dan monetary.

- ➔ Custscale digunakan dalam proses normalisasi dimana nilai awal sebelum normalisasi data berisi nilai setiap kolom yang memiliki skala atau range yang berbeda jauh. Oleh karena itu, perlu dilakukan normalisasi sehingga data tersebut memiliki nilai setiap kolom yang tidak terlalu jauh perbedaan skala atau rangenya.

Syntax

```
#clustering
customerscluster <- customers[2:4]
custscale <- as.data.frame(scale(customerscluster ))
```

Hasil

recency	frequency	monetary
-0.33477928	-0.27005438	-0.324528238
0.71034400	0.41010856	0.472859091
0.25776829	-0.61013585	-0.552043645
0.19244809	-0.61013585	-0.493539683
-0.10615857	-0.27005438	0.071998613
-0.92266113	1.09027150	1.748028775
-0.94598978	-0.27005438	0.540030307
-0.98331561	0.07002709	0.709041752
0.90630462	-0.27005438	-0.540342852
0.49105474	-0.27005438	-0.760057731
-0.91332967	-0.61013585	-0.493539683

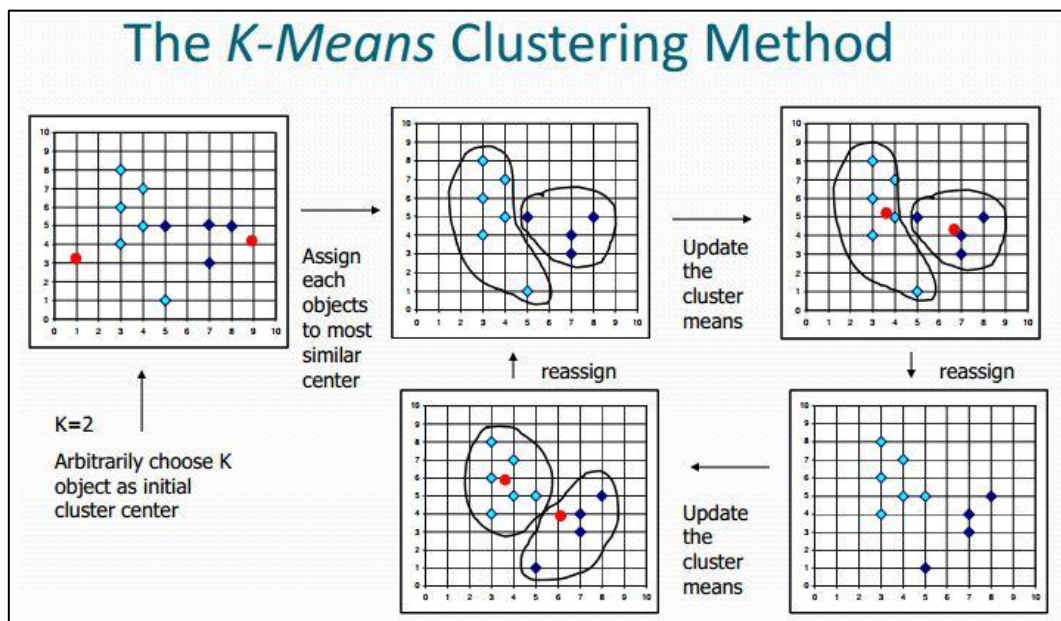


K-MEANS CLUSTERING

K-Means Clustering adalah suatu metode penganalisaan data atau metode Data Mining yang melakukan proses pemodelan tanpa supervisi (unsupervised) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi.

Data clustering menggunakan metode K-Means Clustering ini secara umum dilakukan dengan algoritma dasar sebagai berikut:

1. Tentukan jumlah cluster
2. Alokasikan data ke dalam cluster secara random
3. Hitung centroid/rata-rata dari data yang ada di masing-masing cluster
4. Alokasikan masing-masing data ke centroid/rata-rata terdekat
5. Kembali ke Step 3, apabila masih ada data yang berpindah cluster atau apabila perubahan nilai centroid, ada yang di atas nilai threshold yang ditentukan atau apabila perubahan nilai pada objective function yang digunakan di atas nilai threshold yang ditentukan



Dalam bahasa R, algoritma K-Means Clustering dapat dijalankan dengan fungsi `kmeans()` sebagai berikut.

Syntax

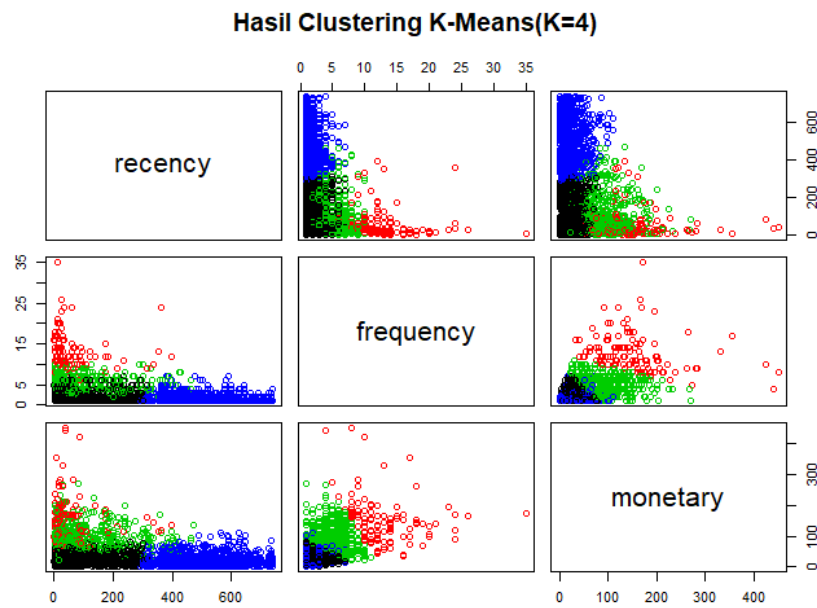
```
#K-means

#k=4
kmeans.result4<- kmeans(custscale,center=4, nstart = 25)
kmeans.result4$cluster
customers$kmeans <- kmeans.result4$cluster
#visualisasi hasil
plot(customers[c("recency", "frequency","monetary")],
      col = kmeans.result4$cluster, main ="Hasil clustering K-Means(K=4)")
#analisis karakteristik kmeans
library(ggplot2)
plot(c(0), xaxt = 'n', ylab = "", type = "l", main="Karakteristik Hasil cluster K-Means(K=4)",
      ylim = c(min(kmeans.result4$centers), max(kmeans.result4$centers)), xlim = c(0, 4))
axis(1, at = c(1:3), labels = names(custscale))
for (i in c(1:4))
  lines(kmeans.result4$centers[i,], lty = i, lwd = 2, col = ifelse(i %in% c(1,3,5),"black", "blue"))
text(x = 0.5, y = kmeans.result4$centers[, 1], labels = paste("cluster", c(1:3)))
```

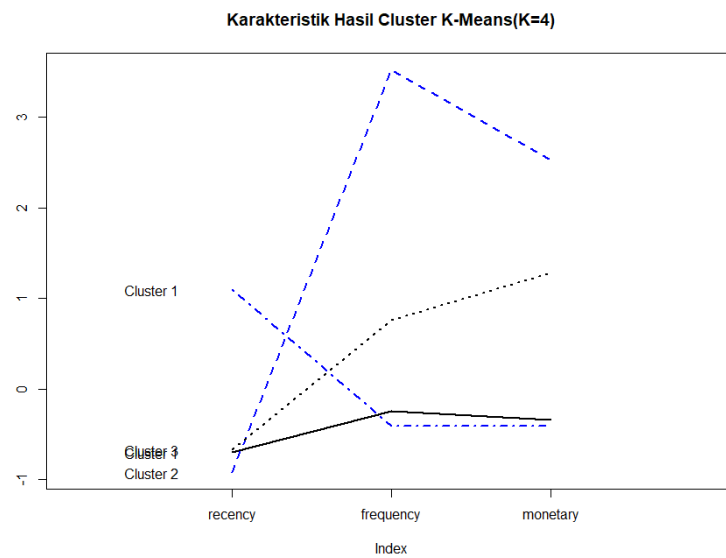


Hasil

Plot Hasil Cluster (Keseluruhan)

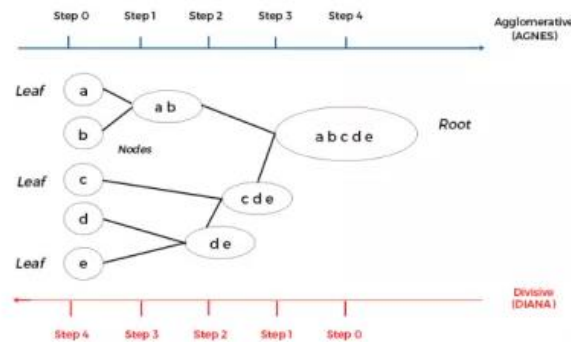


Karakteristik



METODE HIRARKI

Metode Hirarki adalah salah satu metode clustering yang mengelompokkan data berdasarkan urutan tingkat kemiripan dimana data yang mirip akan ditempatkan dalam hirarki yang sama begitupun sebaliknya. Di bawah ini merupakan 2 jenis metode hirarki yaitu Agglomerative dan Divisive Clustering.



Langkah-langkah melakukan Hierarchical clustering yaitu :

1. Melakukan identifikasi terhadap obyek dengan jarak terdekat
2. Menggabungkan obyek ke dalam 1 cluster
3. Menghitung jarak dari satu cluster ke cluster yang lainnya
4. Mengulangi dari awal sampai semua obyek terhubung

Agglomerative Clustering

Agglomerative Clustering adalah suatu metode hirarki yang mengelompokkan data dari N jumlah cluster menjadi 1 cluster. Setiap obyek terdiri dari banyak cluster yang memiliki kemiripan digabung menjadi 1 cluster besar (proses pemusatan cluster). Metode Agglomerative Clustering memiliki beberapa method yang digunakan dalam menghitung tingkat kemiripan. Dalam hal ini terdapat 4 penerapan metode dari Agglomerative Clustering yaitu :

1. Single Linkage

Melakukan clustering dengan berdasarkan pada jarak terpendek . Jika terdapat beberapa obyek yang memiliki jarak hamper mirip atau dekat maka akan dikelompokkan menjadi 1 cluster.

2. Complete Linkage

Melakukan clustering dengan berdasarkan pada jarak terjauh. Jika terdapat bberapa obyek data dengan perbedaan jarak yang jauh maka akan dijadikan 1 cluster.

3. Average Linkage

Melakukan clustering dengan berdasarkan rata-rata jarak dari seluruh obyek data yang digunakan.

4. Ward's Method

Melakukan clustering dengan melakukan berdasarkan dari sum of square error (SSE) .



Syntax

```
#Aglo clustering
library(cluster)
library(factoextra)
#method can be 'average', 'single', 'complete', 'ward'
# Agglomerative Nesting (Hierarchical Clustering)
datacluster.agnes <- agnes(x=customers[2:4], # data matrix
                           stand = TRUE, # standarize the data
                           metric= "euclidian", # metric for distance matrix
                           method="ward" # Linkage method
)

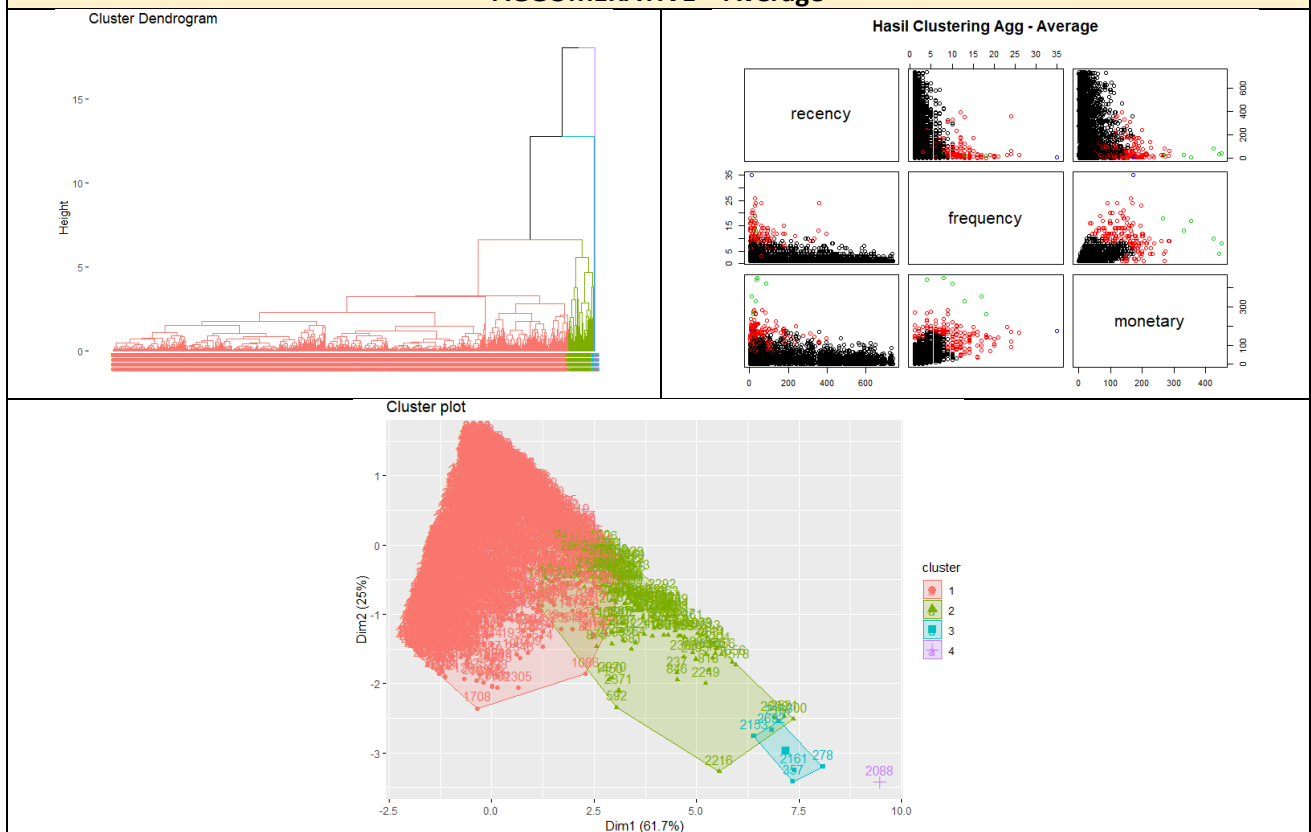
fviz_dend(datacluster.agnes, cex=0.6, k=3)

clust<- cutree(datacluster.agnes, k=3)
fviz_cluster(list(data = customers[2:4], cluster=clust))
```

- ➔ Pada penerapan agglomerative clustering dilakukan percobaan dengan menggunakan 4 method dimana syntax pada bagian method diganti sesuai dengan apa yang akan dicari.
- ➔ Hasil di bawah ini merupakan percobaan dari menggunakan 4 jenis method yang ada di Agglomerative Clustering.

Hasil

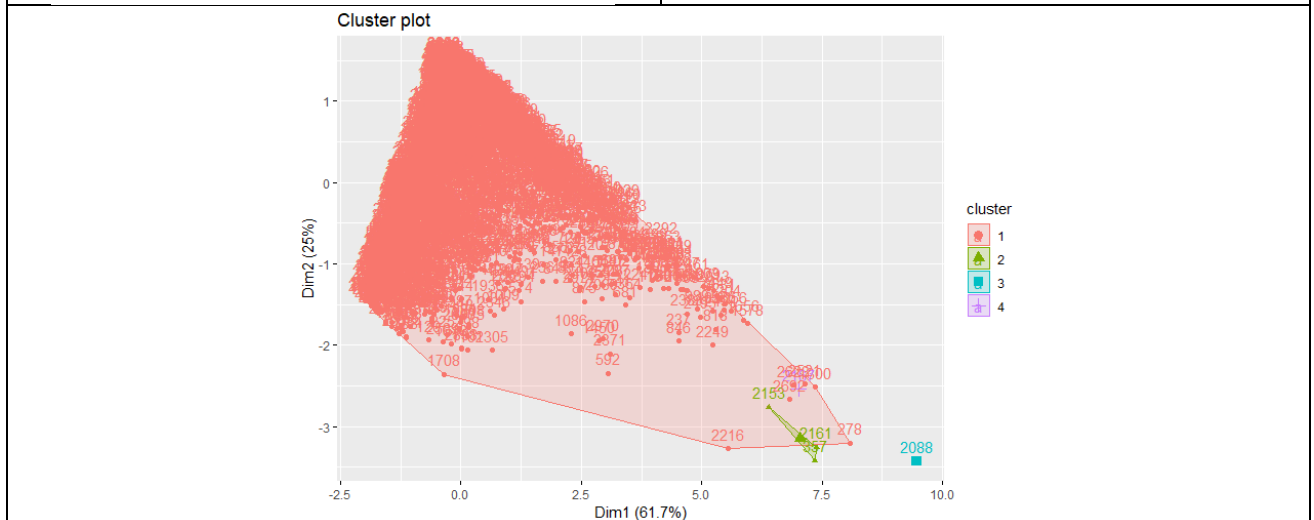
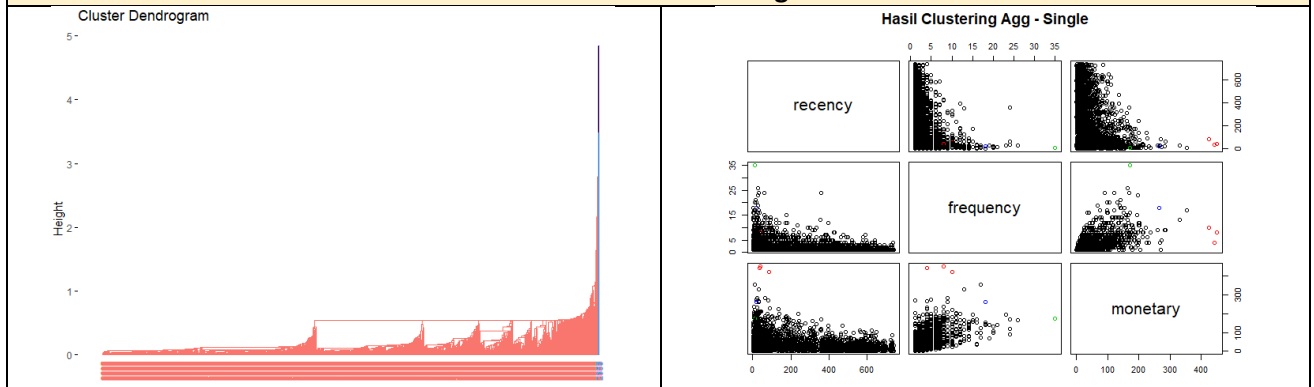
AGGOMERATIVE – Average



- ➔ Percobaan pertama menggunakan method=Average dimana pengelompokkan didasarkan terhadap nilai rata-rata dari seluruh data tersebut.
- ➔ Pada gambar cluster pot di atas, terdapat 4 cluster dimana cluster 4 tidak memiliki anggota lain dalam clusternya karena berada di jarak yang jauh dibandingkan dengan titik obyek yang lainnya. Selain itu, pada penerapan method ini terlihat bahwa terdapat cluster yang tumpang tindih yaitu cluster 1,2,dan 3.

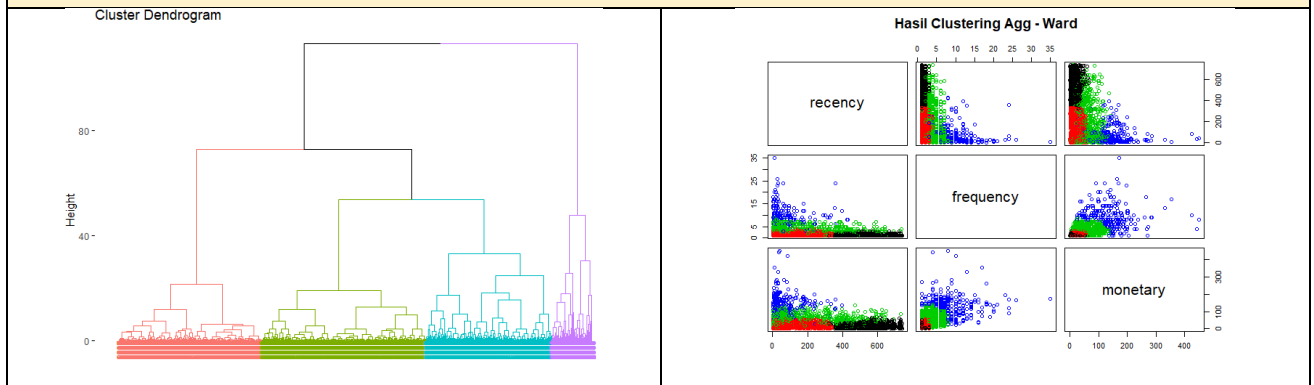


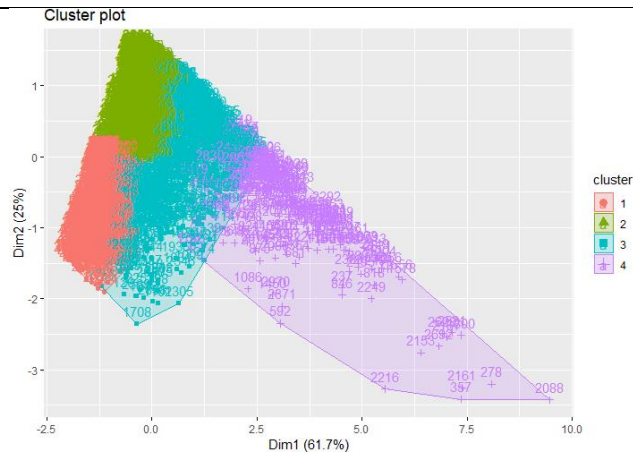
AGGOMERATIVE – Single



- ➔ Percobaan kedua dengan menggunakan method = Single dimana membuat cluster berdasarkan jarak terdekat dari antar obyek dalam data.
- ➔ Dalam penerapan ini terdapat 1 cluster yang memiliki hanya 1 anggota dan ada tumpang tindih antara cluster 1, 2, dan 4. Cluster dengan anggota paling banyak dimiliki oleh cluster 1.

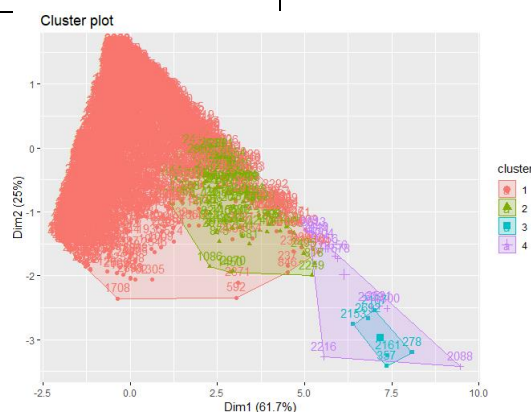
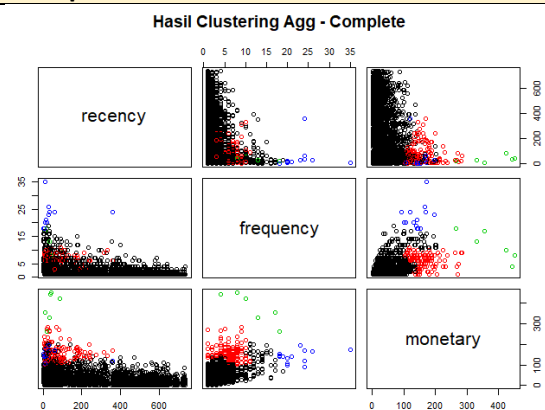
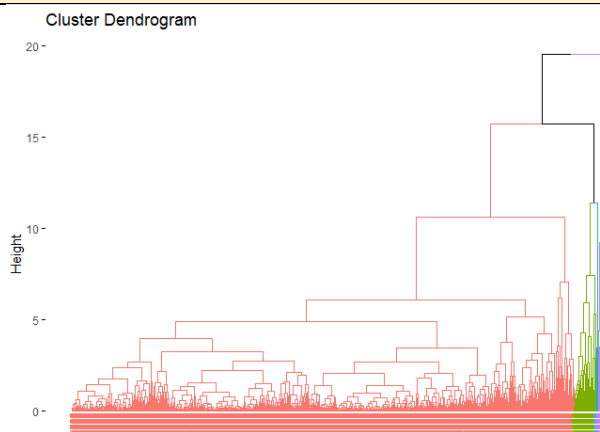
AGGOMERATIVE – Ward





- ➔ Percobaan ketiga dengan menggunakan method = Ward dimana membuat cluster berdasarkan pada nilai Sum of Square Error (SSE).
- ➔ Dalam penerapan method ini dapat dihasilkan tampilan dari 4 cluster yang jelas dimana peristiwa tumpang tindih antar cluster masih tidak terlalu terlihat. Perbedaan dan persebaran warna dapat melihat sekilas bentuk dan persebaran dari clustering. Pada method ini visualiasi dari 4 cluster memiliki kejelasan dalam pemisahan obyek pada data tersebut.

AGGOMERATIVE – Complete



- ➔ Percobaan terakhir dengan menggunakan method = Complete dimana membuat cluster berdasarkan jarak terjauh dari antar obyek.
- ➔ Dalam penerapan ini terdapat 4 cluster yang semuanya tumpang tindih antara cluster satu dengan yang lain. Pada gambar Cluster plot di atas terlihat bahwa cluster 3 dan 4 saling tumpang tindih, cluster 1,2, dan 3 juga mengalami tumpang tindih sehingga penyebaran cluster masih tidak jelas terlihat dengan berdasarkan cluster plot.



Divisive Clustering

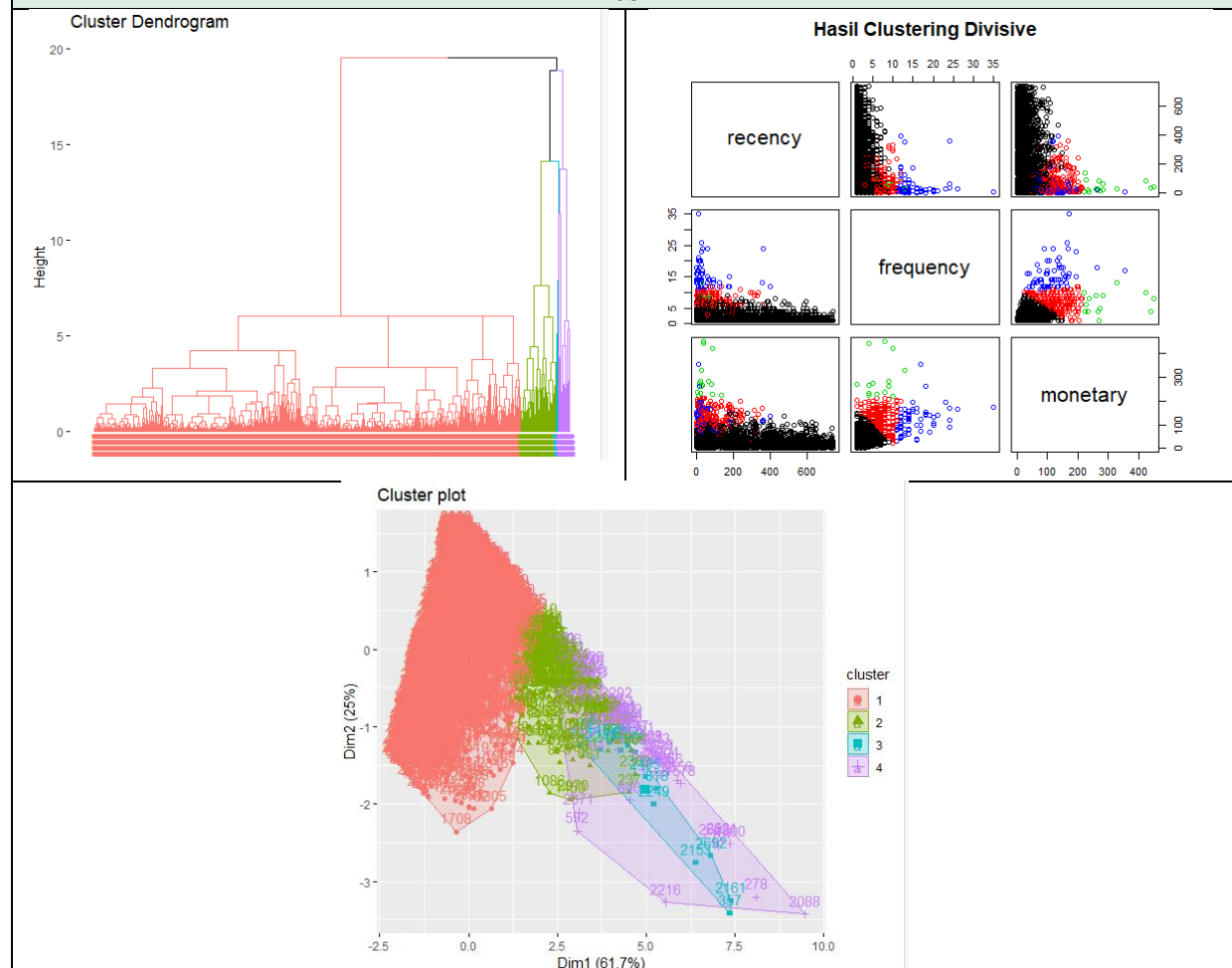
Divisive Clustering adalah suatu metode hirarki yang berkebalikan dengan Agglomerative dimana mengelompokkan suatu data dari 1 cluster menjadi N cluster. Pemisahan atau penyebaran dari 1 cluster besar menjadi beberapa cluster kecil.

Syntax

```
#Divisive clustering
datacluster.diana <- diana(x=custscale, # data matrix
                           stand = TRUE, # standarize the data
                           metric= "euclidian" # metric for distance matrix
)

fviz_dend(datacluster.diana, cex=0.6, k=4)
clustdvs<- cutree(datacluster.diana, k=4)
customers$divisive <- clust.d divisive
plot(customers[c("recency", "frequency", "monetary")],
      col = customers$divisive, main = "Hasil Clustering Divisive")
fviz_cluster(list(data = custscale, cluster=clustdvs))
```

Hasil



- ➔ Percobaan pada Divisive dihasilkan cluster yang saling tumpang tindih dan pada syntax tidak diterapkan method seperti yang dilakukan pada Agglomerative. Penyebaran data memang terlihat dari perbedaan warna namun masih ada tumpang tindih. Hal ini akan membuat bingung dalam melakukan clustering karena bisa jadi 1 obyek data dapat masuk ke dalam anggota cluster yang lain.

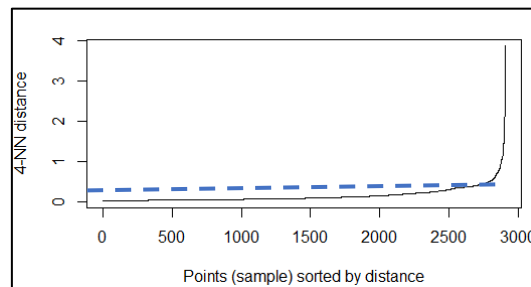


METODE BERBASIS DENSITAS (DBSCAN)

Density-Based Spatial Clustering of Application with Noise (DBSCAN) merupakan sebuah metode clustering yang membangun area berdasarkan kepadatan yang terkoneksi (density connected). DBSCAN merupakan algoritma yang didesain oleh Ester et al pada tahun 1996 dapat mengidentifikasi kelompok-kelompok dalam kumpulan data spasial yang besar dengan melihat kepadatan lokal dari elemen-elemen basis data, dengan hanya menggunakan satu parameter input. DBSCAN juga dapat menentukan apakah informasi diklasifikasikan sebagai noise atau outlier. Disamping itu, proses kerja DBSCAN cepat dan sangat baik untuk berbagai macam ukuran database - hampir linear.

Dalam menerapkan algoritma DBSCAN ada 2 parameter yang dapat dioptimalkan, yaitu *epsilon* (eps) dan *minpts*. Epsilon adalah parameter yang menentukan radius poin yang digunakan untuk menentukan cluster. Sedangkan, parameter minpts digunakan untuk mengatur total poin minimal yang ada pada suatu cluster.

Untuk mengetahui berapa nilai eps yang optimal maka dapat memanggil function `knndisplot()` pada library `dbscan`, dan dihasilkan plot sebagai berikut :

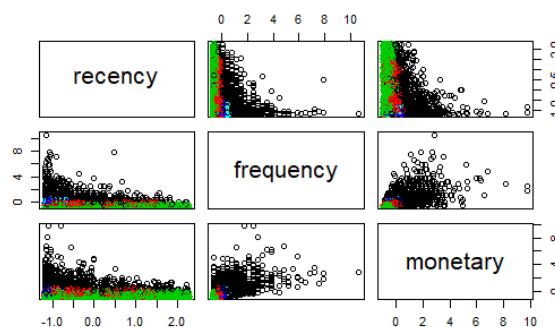


Dapat diketahui dari plot diatas bahwa plot menyudut di distance sekitar 0,3 sehingga nilai parameter eps yang digunakan adalah 0,3.

Syntax

```
#Density Based
library(fpc)
datacluster.ds <- fpc::dbscan(custscale, eps=0.3, MinPts=30)
datacluster.ds$cluster
#Plot cluster
plot(datacluster.ds, custscale)
```

Hasil

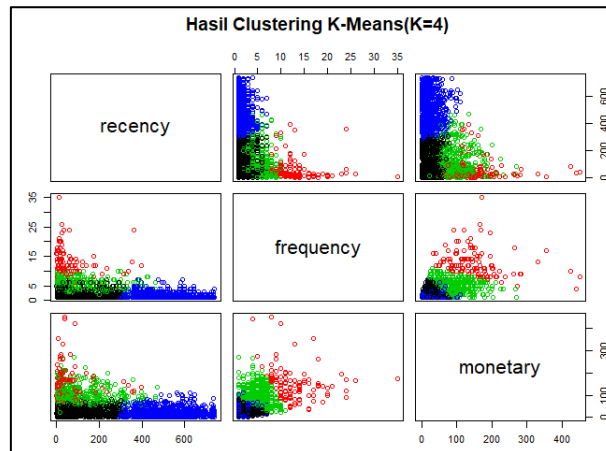


Penjelasan lebih lanjut tentang hasil clustering akan dijelaskan pada bab IV.



PERBANDINGAN HASIL CLUSTERING

Kmeans



➔ Pada gambar scatter plot dari Kmeans dapat terlihat bahwa dalam setiap kategori RFM ada 4 cluster dimana pembagian clusternya juga tampak berbeda.

a. Pembagian cluster antara **Recency dan Frequency**

1. Recency yang memiliki nilai kecil < 400 dimana dapat mengartikan transaksi yang dilakukan customer terbaru akan masuk ke dalam cluster yang memiliki frequency < 10 sehingga membentuk 1 cluster berwarna hitam.
2. Frequency yang memiliki nilai kecil namun recency > 400 maka akan dijadikan cluster tersendiri juga dengan warna biru.
3. Cluster yang berwarna hijau merupakan cluster yang memiliki recency < 600 dengan frequency < 10 namun lebih luas persebarannya dari cluster yang berwarna hitam.
4. Cluster yang berwarna merah merupakan cluster yang memiliki recency antara 200 – 600 lebih dengan frequency yang lebih besar dibandingkan dengan cluster yang lainnya.

b. Pembagian cluster antara **Recency dan Monetary**

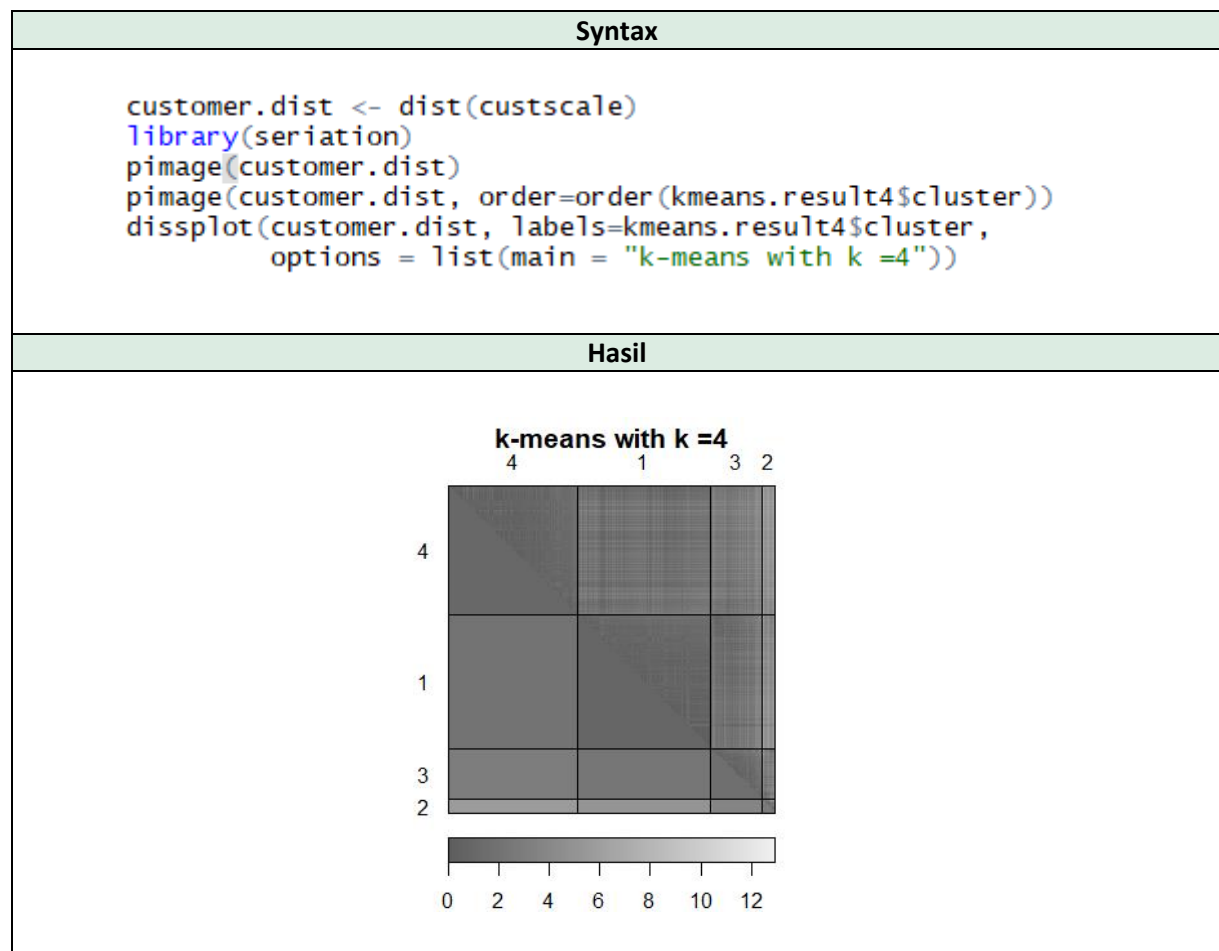
Pembagian cluster dalam hal ini hampir sama dengan pembagian antara recency dan frequency namun terlihat perbedaan antara cluster yang berwarna hijau dan merah. Kedua cluster ini tumpang tindih dengan recency yang < 400 dan monetary persebarannya merata namun cluster berwarna merah memiliki nilai monetary yang lebih banyak yaitu > 300.

c. Pembagian cluster antara **Frequency dan Monetary**

Cluster yang berwarna hitam dan biru terlihat jelas ada tumpang tindih dimana nilai frequency cenderung kecil dan nilai monetary < 200. Sedangkan pada cluster berwarna hijau lebih besar nilai frequency dan monetarynya dibandingkan dengan 2 cluster sebelumnya. Cluster berwarna merah merupakan cluster yang memiliki nilai monetary dan frequency yang menyebar baik nilai yang kecil ataupun besar. Namun, yang banyak memiliki anggota berada di nilai monetary < 300 dan frequency < 25.



Selain dengan menggunakan scatter plot, karakteristik tiap cluster dapat dilihat dengan menggunakan dissimilarity plot dengan memanggil function `dissplot()` pada library `seriation`.

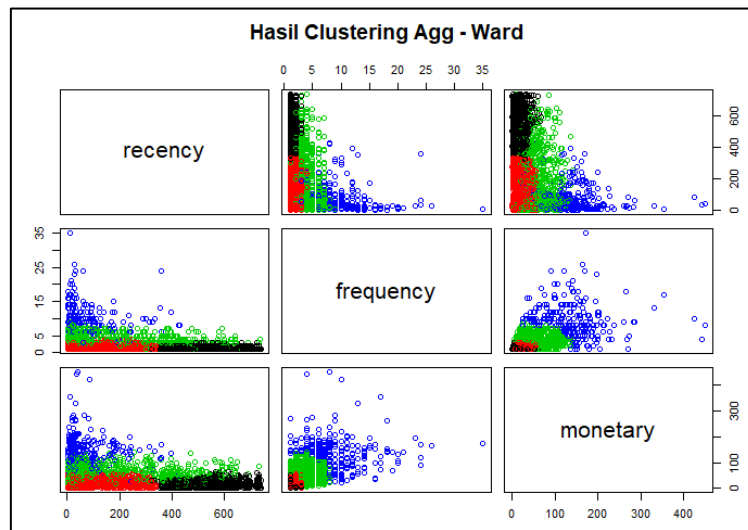


Dari dissimilarity plot yang terbentuk, dapat diketahui bahwa :

1. Cluster 1 dan 2 memiliki tingkat kemiripan yang kecil, ditunjukkan dengan kotak berwarna abu muda.
2. Cluster 1 dan 3 memiliki tingkat kemiripan yang relatif sedang, ditunjukkan dengan kotak berwarna abu terang cenderung gelap.
3. Cluster 1 dan 4 memiliki tingkat kemiripan yang relatif cukup tinggi karena warna kotak yang berwarna abu-abu gelap.
4. Cluster 2 dan 3 memiliki tingkat kemiripan yang relatif sedang karena warna kotak berwarna abu terang cenderung gelap.
5. Cluster 2 dan 4 memiliki tingkat kemiripan yang relatif kecil karena warna kotak berwarna abu terang.
6. Cluster 3 dan 4 memiliki tingkat kemiripan yang relatif sedang karena warna kotak berwarna abu terang cenderung gelap.



Hierarchical Clustering (Agglomerative Clustering-Ward)

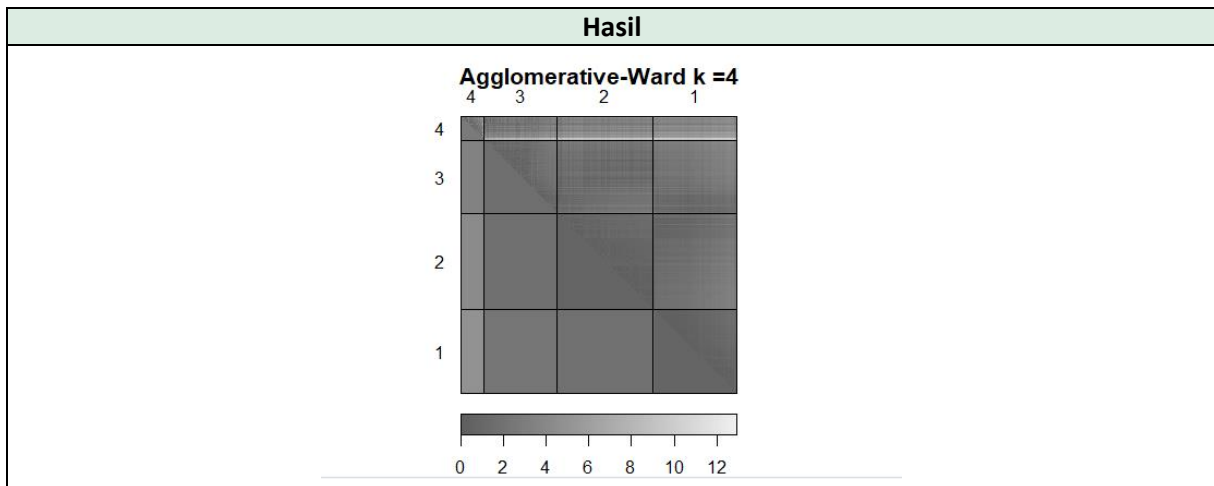


- ➔ Pada metode hirarki yang memiliki bentuk scatter plot bagus dan terlihat warna dari 4 cluster yaitu Agglomerative Clustering-Ward. Hal ini dapat dipahami dari konsep pengambilan SSE yang dilakukan oleh metode Ward dalam mengelompokkan setiap clusternya.
- ➔ Dari hasil scatter plot diatas terlihat bahwa terdapat 4 cluster dengan warna yang berbeda pada setiap kategori RFM.
 - a. Kategori untuk hubungan antara **Recency dan Frequency** hampir sama dengan hasil Kmeans. Perbedaannya terlihat pada nilai cluster yang berwarna hijau recencynya meyebar yaitu antara 0-600 lebih. Sedangkan pada cluster 3 Kmeans frequency yang dikelompokkan nilainya < 600.
 - b. Hubungan antara pengelompokkan **Recency dan Monetary** juga hampir sama persebarannya dengan hasil Kmeans. Perbedaan juga terlihat pada rentang nilai pada cluster hijau mengikuti dari kategori Recency dan Frequency.
 - c. Hubungan antara pengelompokkan **Frequency dan Monetary** terlihat agak mirip dengan perbedaan warna pada pembagian cluster yang di Kmeans. Cluster berwarna merah dan hitam memiliki tumpang tindih dengan nilai frequency dan monetary yang kecil. Sedangkan cluster berwarna hijau memiliki nilai yang lebih besar di kedua sisi. Pada cluster berwarna biru terlihat nilai kedua ketagori cenderung menyebar dengan nilai yang lebih besar dibandingkan dengan ketiga cluster sebelumnya.

Syntax

```
customer.dist <- dist(custscale)
library(seriation)
pimage(customer.dist)
pimage(customer.dist, order=order(customers$Awr d))
displot(customer.dist, labels=(customers$Awr d),
         options = list(main = "Agglomerative-ward k =4"))
```

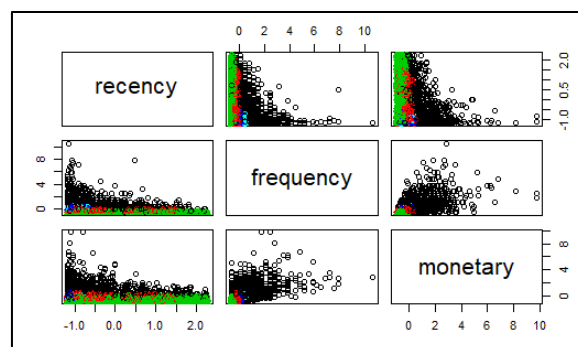




Dari dissimilarity plot yang terbentuk, dapat diketahui bahwa :

1. Cluster 1 dan 2 memiliki tingkat kemiripan yang sedang, ditunjukkan dengan kotak berwarna abu terang cenderung gelap.
2. Cluster 1 dan 3 memiliki tingkat kemiripan yang relatif sedang, ditunjukkan dengan kotak berwarna abu terang cenderung gelap.
3. Cluster 1 dan 4 memiliki tingkat kemiripan yang relatif kecil karena warna kotak berwarna abu terang.
4. Cluster 2 dan 3 memiliki tingkat kemiripan yang relatif tinggi karena warna kotak berwarna abu gelap.
5. Cluster 2 dan 4 memiliki tingkat kemiripan yang relatif kecil karena warna kotak berwarna abu terang.
6. Cluster 3 dan 4 memiliki tingkat kemiripan yang relatif sedang karena warna kotak berwarna abu terang cenderung gelap.

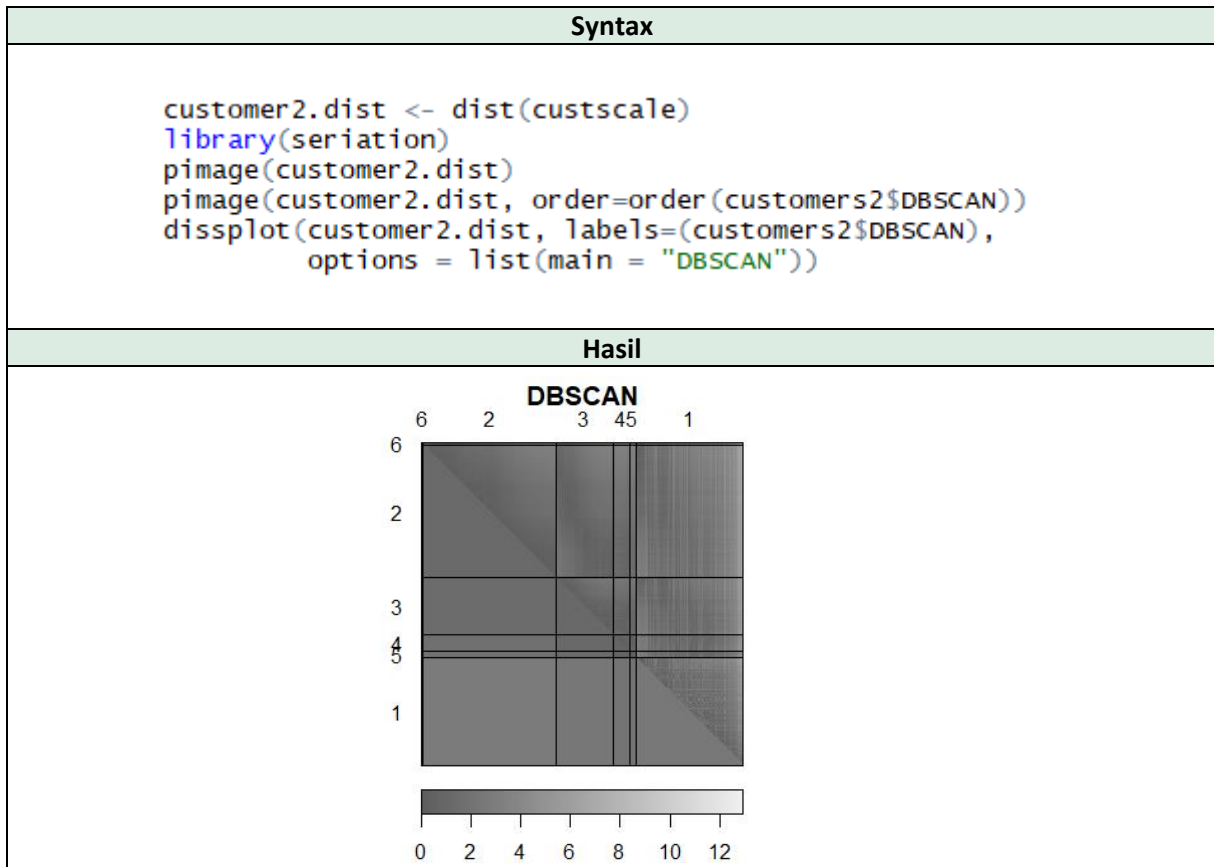
DBSCAN



- ➔ Pada hasil cluster yang terlihat dari scatter plot DBSCAN dapat disimpulkan bahwa banyak cluster yang tumpang tindih dan tidak jelas pembagian cluster dibandingkan dengan metode agglomerative clustering-ward. Cluster berwarna hitam lebih banyak anggotanya karena terdeteksi sebagai outlier.
- a. Hubungan antara Recency dan Frequency terlihat bahwa nilai recency menyebar dari kecil ke besar dan memiliki frequency yang kecil maka akan dijadikan dalam 1 cluster. Jadi dapat disimpulkan bahwa penentuan cluster berdasarkan warna yaitu mengikuti besar kecilnya frequency. Cluster berwarna hitam memiliki jumlah frequency yang cenderung lebih besar namun juga terdapat tumpang tindih di nilai frequency yang kecil.



- b. Hubungan antara pengelompokkan kategori recency dan monetary yaitu hampir mirip dengan recency dan frequency.
- c. Hubungan antara Frequency dan Monetary dapat terlihat bahwa anatra cluster berwarna merah, hijau, dan biru tidak jelas batasan lingkup clusternya. Anggotanya bisa saja saling redundan dimana dapat masuk ke dalam lebih dari 1 cluster. Mayoritas cluster berada pada cluster yang berwarna hitam dengan nilai monetary dan frequency yang cenderung lebih tinggi.



- ➔ Dari hasil dissimilarity plot di atas dapat disimpulkan bahwa tingkat kemiripan masing-masing cluster memiliki kecenderungan sedang dan tinggi. Hal ini dapat dilihat karena plot yang terbentuk berwarna abu agak gelap dan abu gelap. Pada kasus ini, hasil dissimilarity plot dapat diartikan bahwa hasil clustering dengan menggunakan DBSCAN jelek karena tidak ada perbedaan yang signifikan dari setiap cluster yang terbentuk.

Maka, dari ketiga hasil scatter plot dalam pemisahan cluster tersebut, dapat dipilih yang metode terbaik dari perbedaan warna dari setiap cluster dari 3 kategori dengan menggunakan metode **K-Means Clustering**. Hal ini ditunjang dengan adanya perhitungan jumlah cluster optimal pada sub-bab selanjutnya.



PERHITUNGAN JUMLAH CLUSTER OPTIMAL

Perhitungan jumlah cluster optimal dapat dilakukan dengan menghitung Total Within Sum of Square dari tiap jumlah cluster yang diujicobakan.

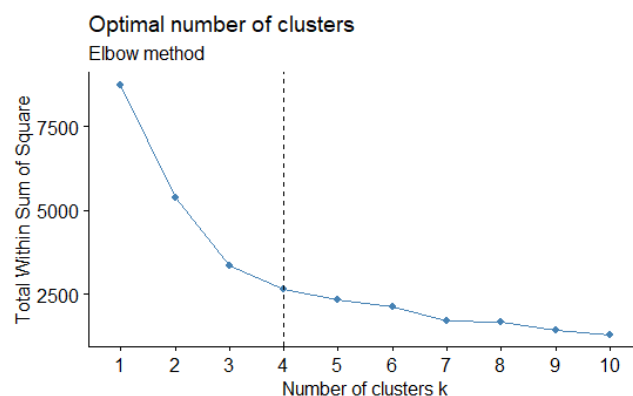
K-MEANS

Pada peneparan Elbow Method dalam perhitungan jumlah kluster optimal dengan menggunakan function Kmeans (FUN=kmeans). Penggunaan method “WSS” digunakan untuk melihat nilai dari SSE. Selain itu, penentuan jumlah kluster optimal dengan melihat gambar grafik yang dihasilkan. Dari grafik tersebut dilihat nilai Total Within Sum of Square yang mengalami penurunan paling drastis.

Syntax

```
fviz_nbclust(custscale, FUN = kmeans, method = "wss") +  
  geom_vline(xintercept = 4, linetype = 2)+  
  labs(subtitle = "Elbow method")
```

Hasil

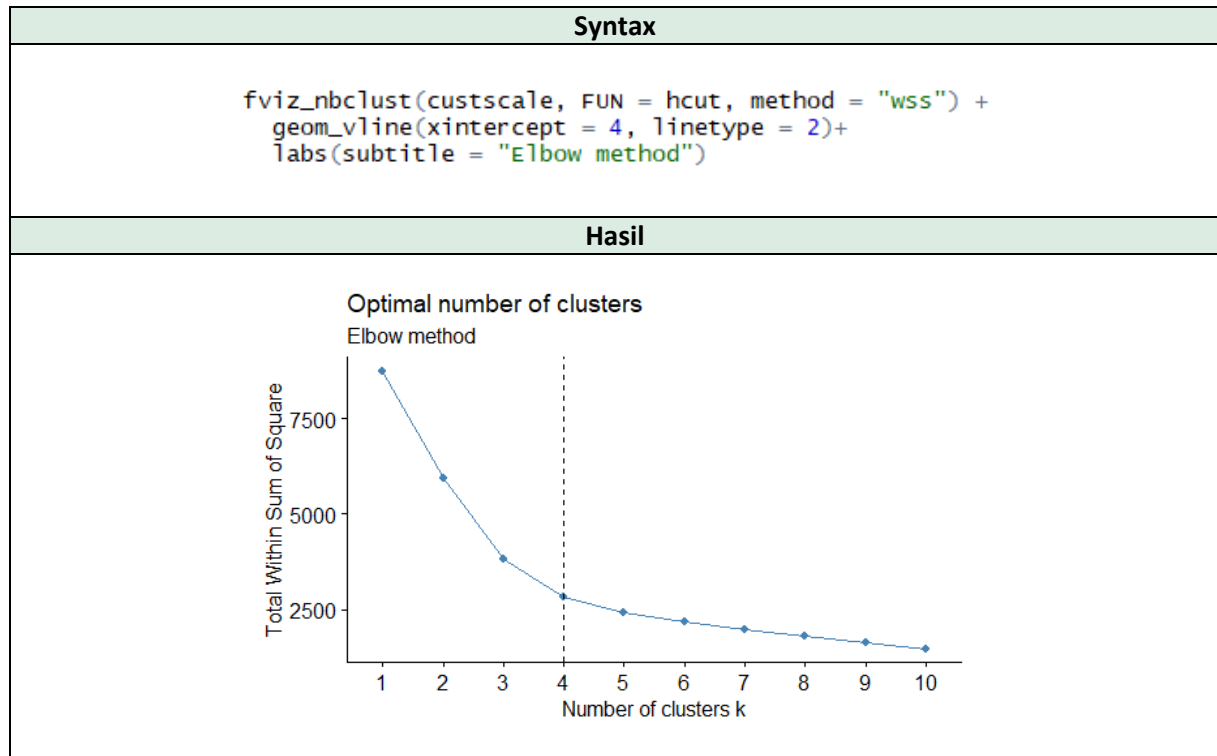


- ➔ Pada hasil grafik di atas, dapat terlihat bahwa penurunan nilai wss yang paling drastic terdapat pada cluster ke-4. Sedangkan untuk cluster lebih dari 5 terlihat penurunan yang cenderung lebih sedikit. Oleh karena itu, cluster yang optimal berada pada cluster ke-4. Pada penempatan titik garis putus-putus pada angka 4 ditentukan dengan menggunakan syntax “xintercept” = 4.



HIERARCICAL CLUSTERING

Pada peneparan Elbow Method dalam perhitungan jumlah kluster optimal dengan menggunakan function hcut (FUN=hcute). Penggunaan method "WSS" digunakan untuk melihat nilai dari SSE. Selain itu, penentuan jumlah kluster optimal dengan melihat gambar grafik yang dihasilkan. Dalam hal ini sama dengan penentuan jumlah cluster optimal di Kmeans dimana dengan melihat grafik dengan nilai Total Within Sum of Square yang mengalami penurunan paling drastis.



Pada pencarian K yang paling optimal dari dua metode yaitu K-Means dan Hierarchical Clustering didapatkan **K = 4**. Pencarian K dilakukan dengan menggunakan Elbow Method. Penerapan Elbow method biasanya digunakan dalam mencari jumlah cluster yang paling optimal pada Kmeans. Namun, pada kasus ini, perhitungan jumlah cluster yang paling optimal dengan menggunakan elbow method juga digunakan dalam metode hirarki. Sedangkan pada DBSCAN belum bisa dianalisis nilai K yang paling optimum, pengoptimalan algoritma DBSCAN dapat dilakukan dengan memperbaiki parameter eps dan minpts seperti yang sudah dijelaskan pada bab 3 bagian DBSCAN.

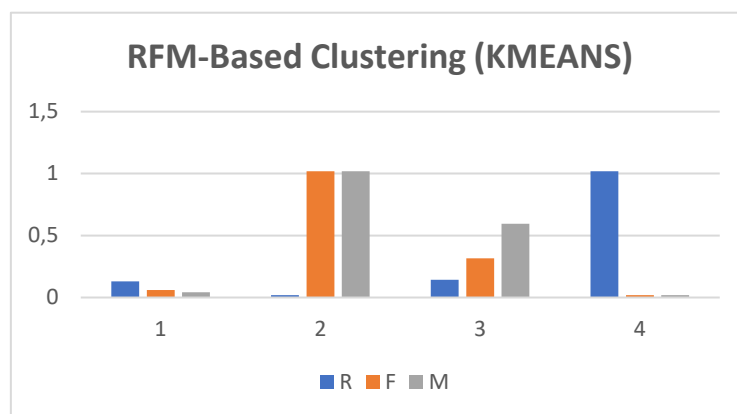
Jika dilihat dari kedua metode yaitu K-Means dan Hierarchical Clustering maka dapat disimpulkan bahwa metode K-Means memiliki nilai SSE lebih kecil dibandingkan dengan metode hirarki. Hal ini dapat dilihat dari titik K = 4 dimana angka Total Within Sum of Square pada metode hirarki lebih besar dengan jarak yang lebih jauh dari 2500. Sedangkan pada K-Means terlihat bahwa jarak antara K=4 dengan angka 2500 sedikit atau lebih kecil dari pada metode hirarki. Mengingat, konsep pada clustering yaitu metode clustering yang terbaik dipilih melalui jumlah cluster yang sedikit dan nilai SSE semakin kecil.



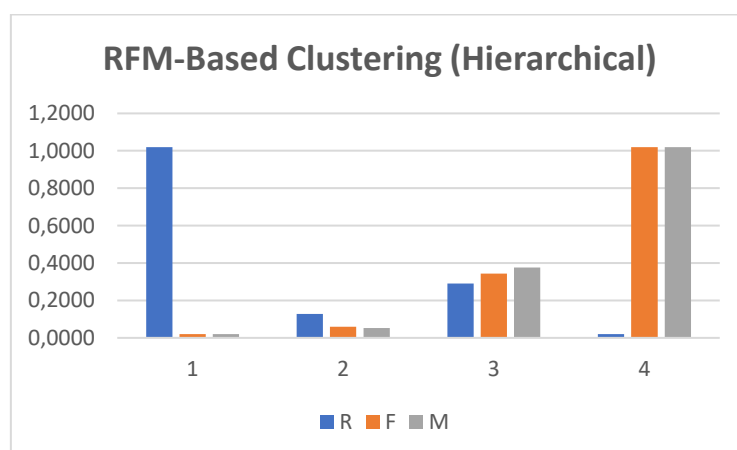
ANALISIS CLUSTER TERHADAP CRM

Pada bagian sebelumnya dibahas mengenai pemilihan jumlah cluster yang paling tepat dimana $K=4$. Bab ini menerapkan dan menghubungkan dari masing-masing cluster ke dalam analisis Customer Relationship Management (CRM) dengan berdasarkan pada Recency, Frequency, dan Monetary (RFM). Di bawah ini merupakan hasil grafik dari tiap cluster dengan berdasarkan RFM.

Recency yang terdapat dalam grafik berbanding terbalik dengan frequency dan monetary. Semakin besar nilai recency maka semakin jelek kualitasnya karena customer melakukan transaksi terakhir pada waktu yang lama. Sedangkan semakin besar nilai monetary dan frequency maka akan semakin bagus kualitasnya karena jumlah transaksi dan uang yang dihabiskan semakin banyak maka peluang untuk menawarkan produk baru dan customer tersebut lebih cenderung memiliki keinginan untuk membeli produk tersebut.



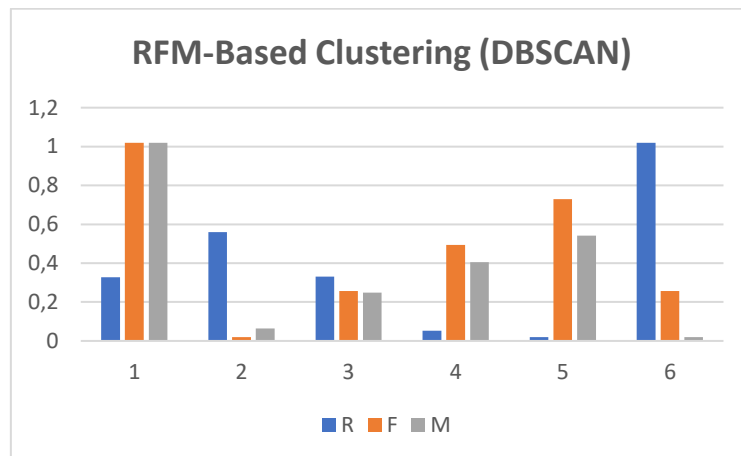
Pada hasil cluster dengan menggunakan metode K-Means dapat terlihat bahwa pelanggan yang menguntungkan bagi perusahaan terdapat pada cluster ke-2. Hal ini dapat diartikan bahwa cluster ke-2 memiliki nilai Recency yang kecil dimana terdapat customer yang melakukan transaksi terbaru dan terdapat customer yang memiliki nilai Frequency dan Monetary yang banyak. Cluster ke-2 memiliki nilai customer yang memiliki jumlah transaksi terbanyak dan menghabiskan uang paling banyak dalam melakukan transaksi.



Pada hasil cluster menggunakan Hierarchical Clustering didapatkan grafik seperti di atas. Dari grafik tersebut dapat disimpulkan bahwa cluster yang terbaik terdapat pada cluster ke-4. Pada cluster ke-4 memiliki nilai Recency yang kecil dimana menandakan transaksi terbaru atau terakhir yang dilakukan oleh customer. Sedangkan jumlah Frequency dan Monetary memiliki nilai yang tinggi. Hal ini dapat



merepresentasikan bahwa terdapat customer yang banyak melakukan transaksi dan menghabiskan jumlah uang yang banyak.



Pada hasil cluster dengan menggunakan DBSCAN dapat terlihat grafik yang paling bagus ditunjukkan pada cluster ke-1. Pada cluster ke-1 nilai Recency mungkin lebih besar dari pada cluster 4 dan 5, namun untuk grafik frequency dan monetary yang memiliki jumlah terbanyak berada pada cluster 1. Untuk cluster terbaik yang kedua dan ketiga yaitu terdapat apda cluster 5 dan 6. Pada cluster 5 memiliki jumlah nilai recency yang kecil. Sedangkan untuk frequency lebih besar dibandingkan dengan monetary. Pada cluster ke-4 juga sama dengan cluster 5 dimana rentang nilainya lebih besar atau banyak di cluster 5.



KESIMPULAN

Dari hasil clustering ketiga metode tersebut, dapat disimpulkan adanya kurang lebih 4 penggolongan customer, sebagai berikut :

Cluster	Karakteristik			Interpretasi
	Recency	Frequency	Monetary	
1	Rendah	Tinggi	Tinggi	Loyal & Profitable Customer Pelanggan ada cluster ini adalah pelanggan yang paling setia karena sangat sering melakukan transaksi dan juga pembelian yang dilakukan dalam skala yang besar.
2	Sedikit Rendah	Sedikit Tinggi	Sedikit Tinggi	Valuable Customer Pelanggan pada cluster ini adalah pelanggan yang baru-baru ini melakukan transaksi dengan frekuensi pembelian dan skala pembelian yang sedikit tinggi.
3	Sedikit Tinggi	Sedikit Rendah	Sedikit Rendah	First Time Customer Pelanggan pada cluster ini adalah pelanggan yang melakukan pembelian tidak terlalu sering juga dalam skala pembelian yang kecil. Selain itu, pelanggan pada golongan ini juga sudah cukup lama tidak melakukan transaksi lagi.
4	Tinggi	Rendah	Rendah	Churn Customer Pelanggan pada klaster ini adalah pelanggan yang paling buruk karena dari ketiga atribut (R, F dan M) tidak ada yang baik. Pelanggan sudah lama dan sangat jarang melakukan transaksi, dan juga saat melakukan transaksi skalanya sangat kecil.

SARAN

Pada tugas penggalian data ini proses identifikasi rekomendasi tidak dilakukan. Hal ini karena pada tugas ini berfokus terhadap analisis dna penentuan clustering dengan berdasarkan kategori RFM, sedangkan untuk selanjutnya melakukan identifikasi rekomendasi merupakan bidang ilmu yang lain yaitu yang lebih mendalam tentang ilmu bisnis dan CRM karena berkaitan dengan kelangsungan proses bisnis tersebut. Maka dari itu, dari hasil analisis yang dilakukan, dapat dilanjutkan dengan melakukan identifikasi rekomendasi untuk tiap segementasi pelanggan agar perusahaan dapat meningkatkan keuntungan dan melakukan upaya-upaya CRM yang tepat sasaran.



BAB VI : REFERENSI

- <https://www.technosoft.co.id/2018/08/16/customer-relationship-management-crm-terbaik/>
- <https://www.hestanto.web.id/analisis-rfm-pada-pemasaran-online/>
- http://repository.its.ac.id/70958/1/5211100102-Undergraduate_Thesis.pdf

