

# **Analisis Transaksi Penjualan E-Commerce Untuk Mengoptimalkan Promosi Produk**



Disusun oleh:  
**Mohamad Irwan Afandi**

**Februari 2021**

## LATAR BELAKANG

Tahun 2020 merupakan tahun yang sangat berat bagi semua pelaku bisnis, dimana pandemic COVID-19 telah merubah hampir sebagian besar alur bisnis, kebiasaan *customer* bahkan perekonomian *customer* itu sendiri. Kita tahu sendiri bahwasannya COVID-19 membuat semua kegiatan semakin dibatasi dengan penerapan PSBB, *social distancing* dan himbauan untuk tetap dirumah jika tidak memiliki keperluan. Tujuannya cukup sederhana, yaitu untuk memutuskan rantai persebaran covid sehingga semua kegiatan bisa kembali seperti sedia kala. Namun apa imbas dari semua itu? Banyak bisnis yang belum terdigitalisasi menjadi bangkrut, hal ini karena sebagian besar orang memilih berbelanja secara online karena dianggap lebih aman. Selain itu banyaknya PHK atau pemotongan gaji karyawan yang terjadi karena keuangan perusahaan pun juga tidak stabil. Hal ini menyebabkan orang-orang lebih memilih berhemat sehingga bisa dikatakan tingkat konsumtif masyarakat menurun selama pandemic ini.

Terlepas dari semua itu, industri *e-commerce* merupakan salah satu sektor yang tidak terlalu terkena dampak dari COVID-19. Mungkin memang minat masyarakat untuk berbelanja menurun, tetapi hal ini juga bisa ditutupi dengan banyaknya masyarakat yang awalnya berbelanja secara tradisional (COD) berubah menjadi online. Peluang ini tentunya harus bisa dioptimalkan oleh industri *e-commerce* untuk meningkatkan jumlah pembeli atau menjaga pembeli sebelumnya agar tetap berbelanja menggunakan *platform* mereka. Salah satu caranya adalah dengan memberikan promo menarik bagi para pelanggannya. Namun yang menjadi permasalahan adalah bagaimana cara memberikan promo yang efektif dimana promo tersebut diharapkan mampu meningkatkan jumlah transaksi? Ada beberapa cara yang bisa dilakukan yaitu dengan melakukan ***customer segmentation, time analysis dan product analysis***.

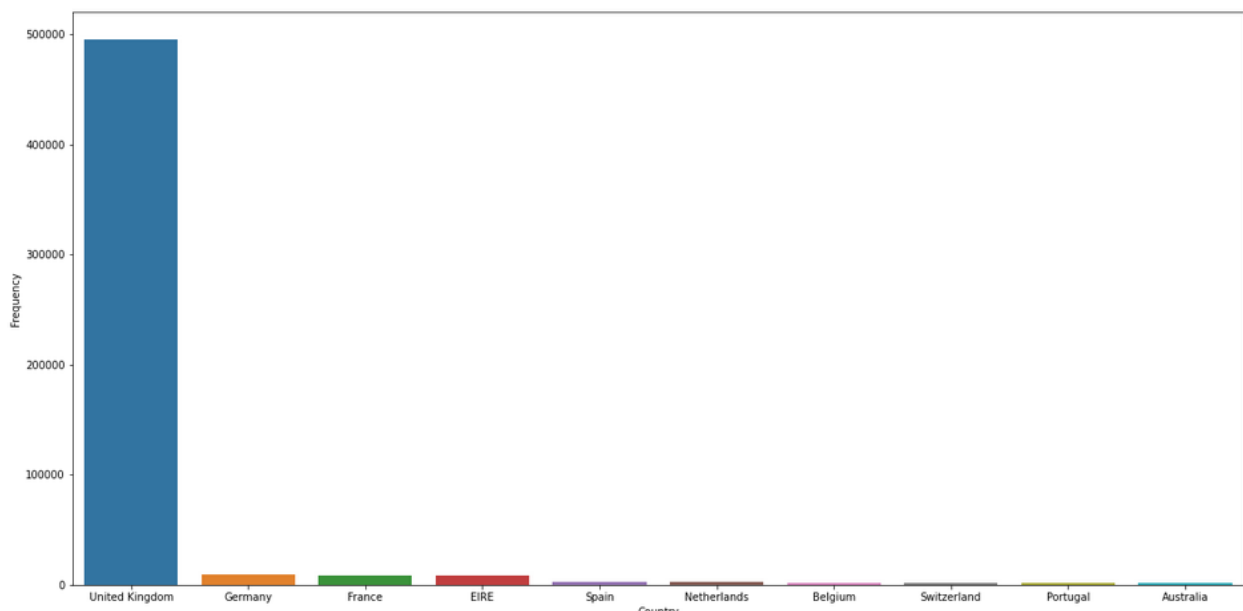
Berdasarkan *case study* yang diberikan, kita diminta untuk melakukan analisis pada data transaksi penjualan dengan harapan mendapatkan *insight* dari data tersebut. Kita tidak bisa langsung melakukan analisis karena data yang diberikan masih berupa *raw of data* sehingga perlu dilakukan beberapa tahapan *preprocessing* terlebih dahulu untuk menghasilkan data yang lebih bersih. Adapun langkah-langkah yang akan dikerjakan pada proses analisis ini adalah *preprocessing (data understanding, missing value handling and anomaly data handling), customer segmentation, transaction analysis based on time series and product analysis*.

## PREPROCESSING

*Pertama* yang kita lakukan adalah *data understanding*, kita akan cari tahu seperti apa datanya dan masalah apa saja yang ada pada data tersebut. Hal pertama yang bisa kita lakukan adalah mengecek info datasetnya terlebih dahulu.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   InvoiceNo        541909 non-null object
1   StockCode       541909 non-null object
2   Description      540455 non-null object
3   Quantity        541909 non-null int64
4   InvoiceDate      541909 non-null object
5   UnitPrice       541909 non-null float64
6   CustomerID      406829 non-null float64
7   Country         541909 non-null object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

Berdasarkan informasi tersebut kita bisa melihat bahwa *dataset* tersebut memiliki 8 fitur, dimana terdapat 2 fitur yang memiliki *missing value* yaitu *Description* dan *CustomerID*. Kemudian ada fitur *Country*, dimana dengan fitur ini kita bisa melihat semua transaksi berdasarkan negaranya. Nah kita coba cek persebaran data transaksi yang kita miliki berdasarkan masing-masing negaranya, dan hasilnya bisa dilihat di bawah ini.



Ternyata transaksi yang paling dominan terjadi di **United Kingdom**, oleh sebab itu kita harus memfilter data transaksi tersebut dan memilih transaksi dari *United Kingdom* saja. Hal ini dilakukan agar hasil analisis kita semakin baik, karena setiap negara pasti memiliki *behavior customer* yang berbeda-beda.

**Kedua** sesuai dengan permintaan soal, kita diminta untuk merubah tahun transaksi dari 2010-2011 menjadi 2020-2021. Nah yang perlu digaris bawahi adalah pada proses ini kita tidak ingin kehilangan informasi hari apa transaksi tersebut dilakukan. Hal ini karena kita juga akan melakukan analisa apakah hari juga mempengaruhi minat *customer* dalam berbelanja. Sehingga caranya adalah kita cek kapan transaksi terakhir pada tahun 2011 terjadi, yaitu pada tanggal 09 Desember 2011 dan bertepatan dengan hari Jumat. Maka kita bisa mengubah tanggal transaksi terakhir tersebut ke tanggal 12 Februari 2021 (hari jumat juga), adapun untuk transaksi yang lain akan menyesuaikan.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
541894	581587	22631 CIRCUS PARADE LUNCH BOX	12	2011-12-09 12:50:00	1.95	12680.0

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
541894	581587	22631 CIRCUS PARADE LUNCH BOX	12	2021-02-12 12:50:00	1.95	12680.0

**Langkah ketiga** yaitu *handling missing value*. Terdapat tiga *case* pada tahap ini dimana *case* yang pertama *Description* dan *CustomerID*-nya *null*. Kita harus menghapus semua data transaksi dengan kondisi tersebut karena tidak ada *insight* yang bisa kita dapatkan. Kedua *CustomerID*-nya saja yang *null*, kita tidak bisa menghapus data ini karena bisa jadi transaksi dilakukan memang tanpa *CustomerID* atau terdapat kesalahan sistem. Namun kita tetap bisa menggunakan data tersebut untuk menganalisis produk yang terjual nantinya. *Case* yang ketiga adalah transaksi dengan data *Description*-nya berupa tanda tanya. *Case* ini sebenarnya sama dengan *case* pertama sehingga kita bisa langsung menghapus data tersebut. Contoh kondisi data adalah sebagai berikut.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
622	536414	22139 NaN	56	2020-02-05 11:52:00	0.0	NaN

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
1443	536544	21773 DECORATIVE ROSE BATHROOM BOTTLE	1	2020-02-05 14:32:00	2.51	NaN

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
7313	537032	21275 ?	-30	2020-02-07 16:50:00	0.0	NaN

**Langkah keempat** yaitu *anomaly data handling*. Bagaimana kita bisa tahu terdapat anomali pada data kita? Jawabannya adalah bisa dengan melihat *summary statistic data* atau menggali masing-masing fiturnya satu persatu. Berikut ini hasil dari *summary statistic* dari data transaksi yang kita miliki.

	Quantity	UnitPrice	CustomerID
count	540408.000000	540408.000000	406829.000000
mean	9.606525	4.623921	15287.690570
std	217.889936	96.893831	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

Berdasarkan hasil *summary statistic* di atas kita bisa lihat bahwa *min value* pada *Quantity* dan *Unit Price* itu negatif sedangkan *max* valuenya positif namun sangat besar. Jika dibandingkan dengan nilai dari *quantile* 1, 2 dan 3 memiliki perbedaan yang sangat jauh (*outlier*). Selain itu sepertinya juga tidak mungkin ada data transaksi dengan nilai *Quantity* produk atau *UnitPrice* yang negatif (kecuali *cancel order* atau *discount*). Untuk memastikannya kita akan coba untuk menggali data lebih jauh lagi. Ketika kita coba untuk cek datanya, ternyata ada hal yang sedikit aneh dimana ada *StockCode* yang nilainya berupa angka saja atau kombinasi angka dan huruf. Kemudian untuk data yang nilai *Quantitinya* negatif selalu dengan *InvoiceNo* yang diawali dengan huruf C. Hal ini yang membuat penasaran ada kombinasi apa saja pada *InvoiceNo* dan *StockCode*. Berikut ini hasil pengecekan yang dilakukan pada data transaksi dengan *InvoiceNo* yang diawali dengan huruf C dan *Quantity*-nya lebih dari 0. Ternyata tidak ditemukan data, sehingga benar jika semua transaksi yang *InvoiceNo*-nya diawali dengan huruf C, maka nilainya selalu negatif.

```
df3[(df3.InvoiceNo.str.contains('C')) & (df3.Quantity >= 0)]
```

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
-----------	-----------	-------------	----------	-------------	-----------	------------

Pertama untuk *InvoiceNo*, ternyata ada 3 jenis yaitu data transaksi yang diawali dengan huruf C, huruf A atau hanya kombinasi angka saja. Untuk yang diawali dengan huruf A sepertinya tidak penting dan juga hanya terdapat 3 data, sehingga kita bisa langsung menghapus data tersebut.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
299982	A563185	B	adjust bad debt	1	2020-10-16 14:50:00	11062.06	NaN
299983	A563186	B	adjust bad debt	1	2020-10-16 14:51:00	-11062.06	NaN
299984	A563187	B	adjust bad debt	1	2020-10-16 14:52:00	-11062.06	NaN

Kemudian data yang diawali dengan huruf C, memang semua nilai *Quantity* nya negatif, namun disini kita mendapat *insight* baru yaitu *StockCode* ini memiliki arti tersendiri yang perlu kita cek juga. Berikut hasil dari pengecekan data yang diawali dengan huruf C pada *InvoiceNo* nya.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
141	C536379	D	discount	-1	2020-02-05 09:41:00	27.50	14527.0
154	C536383	35004C	set of 3 coloured flying ducks	-1	2020-02-05 09:49:00	4.65	15311.0
235	C536391	22556	plasters in tin circus parade	-12	2020-02-05 10:24:00	1.65	17548.0
236	C536391	21984	pack of 12 pink paisley tissues	-24	2020-02-05 10:24:00	0.29	17548.0
237	C536391	21983	pack of 12 blue paisley tissues	-24	2020-02-05 10:24:00	0.29	17548.0

Karena ketika dicek *Description* nya ada yang berupa *discount*, berarti asumsi kita semakin benar bahwa data dengan nilai negatif jika bukan *discount*, *cancel order*, atau hal yang berkaitan dengan produk yang lain. Dari sini kita bisa lihat semua deskripsi produk yang **tidak diawali dengan huruf C** dan nilainya negatif. Ternyata hasilnya berkaitan dengan data produk yang rusak.

```

The number of quantity: 151305
check                      123
damaged                    57
damages                    46
sold as set on dotcom      20
unsaleable, destroyed.     9
thrown away                9
??                          7
damages?                   5
ebay                       5
wet damaged                5
smashed                    4
missing                    4
dotcom                     3

```

Karena kita tidak membutuhkan data ini, kita bisa langsung memfilter data tersebut, sehingga data yang rusak tidak masuk kedalam analisis kita.

```

#drop damage data product
df3.drop(index=df3[(~df3.InvoiceNo.str.contains('C')) & (df3.Quantity <= 0)].index, inplace=True)

```

Lalu bagaimana *handling* data yang *InvoiceNo* nya diawali dengan huruf C dan *Quantity*-nya negatif? Apakah benar semuanya berupa *discount* dan *cancel order*? Ternyata tidak semua dan terdapat data anomali dimana *customer* tidak melakukan transaksi pada hari itu namun tiba-tiba terdapat data transaksi dengan nilai *Quantity* negatif pada tanggal tersebut. Sehingga kita harus menghapusnya agar hasil dari proses analisis kita semakin berkualitas. Contoh data anomalnya adalah sebagai berikut.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID
235	C536391	22556	plasters in tin circus parade	-12	2020-02-05 10:24:00	1.65	17548.0
236	C536391	21984	pack of 12 pink paisley tissues	-24	2020-02-05 10:24:00	0.29	17548.0
237	C536391	21983	pack of 12 blue paisley tissues	-24	2020-02-05 10:24:00	0.29	17548.0
238	C536391	21980	pack of 12 red retrospot tissues	-24	2020-02-05 10:24:00	0.29	17548.0
239	C536391	21484	chick grey hot water bottle	-12	2020-02-05 10:24:00	3.45	17548.0
240	C536391	22557	plasters in tin vintage paisley	-12	2020-02-05 10:24:00	1.65	17548.0
241	C536391	22553	plasters in tin skulls	-24	2020-02-05 10:24:00	1.65	17548.0
165024	550755	22585	pack of 6 birdy gift tags	24	2020-06-24 12:01:00	1.25	17548.0
165025	550755	22082	ribbon reel stripes design	10	2020-06-24 12:01:00	1.65	17548.0
165026	550755	22081	ribbon reel flora + fauna	10	2020-06-24 12:01:00	1.65	17548.0
165027	550755	22079	ribbon reel hearts design	10	2020-06-24 12:01:00	1.65	17548.0
165028	550755	22926	ivory giant garden thermometer	4	2020-06-24 12:01:00	5.95	17548.0
177223	C552049	22926	ivory giant garden thermometer	-4	2020-07-10 10:00:00	5.95	17548.0
177224	C552049	22585	pack of 6 birdy gift tags	-24	2020-07-10 10:00:00	1.25	17548.0
177225	C552049	22082	ribbon reel stripes design	-10	2020-07-10 10:00:00	1.65	17548.0

Bisa kita lihat dari hasil *filter* tersebut dimana *customer* dengan ID 17548 tidak melakukan transaksi lain pada tanggal 5 Februari 2020 namun tiba-tiba ada data *cancel order* (data sudah diurutkan berdasarkan tanggal secara *ascending*). Hal semacam ini yang membuat analisis kita nantinya semakin berantakan, sehingga kita perlu menghapus data tersebut. Namun bagaimana cara menghapusnya akan dibahas dalam penerapan *Customer Segmentation*.

Langkah selanjutnya adalah melakukan pengecekan terhadap *StockCode* yang tidak diawali dengan angka. Ternyata ada maksud tersendiri dari *StockCode* tersebut dan berikut ini hasilnya.

```
array(['POST', 'D', 'C2', 'DOT', 'M', 'BANK CHARGES', 'S', 'AMAZONFEE',
      'DCGS0076', 'DCGS0003', 'gift_0001_40', 'DCGS0070', 'm',
      'gift_0001_50', 'gift_0001_30', 'gift_0001_20', 'DCGS0069',
      'DCGSBOY', 'DCGS GIRL', 'gift_0001_10', 'PADS', 'DCGS0004',
      'CRUK'], dtype=object)
```

Contoh maksud dari kode tersebut:

- |                            |  |
|----------------------------|--|
| 1. POST, DOT, C2           | : berhubungan dengan biaya pengiriman        |
| 2. D, cruk                 | : discount (nilai <i>Quantity</i> selalu -1) |
| 3. M/m                     | : transaksi manual                           |
| 4. BANK CHARGES, AMAZONFEE | : biaya yang dikenakan platform              |
| 5. S                       | : data testing saja (input sample)           |
| 6. Gift_                   | : berupa voucher                             |
| 7. DCGS                    | : produk biasa                               |

Nah disini kita harus menghapus data yang *StockCode* nya berupa huruf S, karena nilai dari *Quantity* nya selalu negatif dan juga hanya data percobaan. Setelah proses ini selesai, langkah selanjutnya adalah melakukan *Customer Segmentation*.



## CUSTOMER SEGMENTATION

*Customer Segmentation* merupakan salah satu cara untuk mengelompokkan *customer* menjadi beberapa kelompok berdasarkan kemiripan *characteristic* dari masing-masing *customer* tersebut. Adapun untuk mengelompokkannya kita bisa menggunakan algoritma *unsupervised learning*, salah satunya adalah *KMeans*. Sebelum melakukan proses *clustering*, terlebih dahulu kita harus membuat fitur yang bisa digunakan dalam pengelompokan *customer* tersebut berdasarkan data transaksi yang dimiliki. Dalam *case* ini kita akan menggunakan **RFM method** (Recency, Frequency and Monetary), salah satu metode yang paling umum digunakan pada *case customer segmentation*.

Sebelum membuat fitur baru untuk RFM, **langkah pertama** yang harus kita lakukan adalah mengatasi *cancel order* dan data anomali (langkah terakhir dari tahap *preprocessing*). Cara mengatasi *cancel order* adalah kita akan menambahkan semua transaksi yang pada tanggal dan jam yang sama dikelompokkan berdasarkan *CustomerID* dan *Description*. Misalkan *customer* membeli produk X sebanyak 10, ternyata pada waktu itu juga dicancel sebanyak 8 buah. Nah dengan proses ini sisa ordernya menjadi 2 buah saja. Hal ini bertujuan untuk mengurangi angka negatif pada fitur *Quantity*. Sekarang tinggal data anomali dimana *customer* tidak memesan produk namun tiba-tiba ada transaksi dengan nilai negatif pada *Quantity* transaksinya. Contoh data anomali adalah sebagai berikut.

```
ex_anomali_data = df_customer_group[df_customer_group.CustomerID==15128].sort_values(by='InvoiceDate').reset_index()
ex_anomali_data.loc[30:40]
```

	index	Description	InvoiceDate	UnitPrice	CustomerID	Quantity
30	18503	alarm clock bakelike red	2020-07-08 12:00:00	3.75	15128.0	4
31	34264	black kitchen scales	2020-07-08 12:00:00	8.50	15128.0	2
32	30127	bathroom metal sign	2020-07-08 12:00:00	0.55	15128.0	12
33	253296	postage	2020-07-22 12:00:00	3.71	15128.0	-1
34	264072	red kitchen scales	2020-07-27 17:00:00	8.50	15128.0	-1
35	715	set 2 tea towels i love london	2020-07-28 11:00:00	3.25	15128.0	-6
36	257818	recipe box pantry yellow design	2020-07-28 11:00:00	2.95	15128.0	-1
37	3068	12 pencils tall tube skulls	2020-08-28 11:00:00	0.85	15128.0	12
38	311788	set of 4 knick knack tins london	2020-08-28 11:00:00	4.15	15128.0	6
39	237492	piggy bank retrospot	2020-08-28 11:00:00	2.55	15128.0	4
40	54065	ceramic strawberry cake money bank	2020-08-28 11:00:00	1.45	15128.0	12

Perhatikan data dengan nomor urut 34, dimana tiba-tiba terdapat transaksi pada tanggal 27 Juli 2020 padahal hari sebelumnya atau hari itu tidak dilakukan transaksi sama sekali. Bahkan dari keseluruhan transaksi *customer* tidak pernah memesan produk tersebut. Sehingga semua nilai negatif untuk quantity harus dihapus kecuali data discount dan pengiriman (*description: postage, discount, cruck commission and carriage*).



```
ex_anomali_data[ex_anomali_data.Description == 'red kitchen scales']
```

	index	Description	InvoiceDate	UnitPrice	CustomerID	Quantity
34	264072	red kitchen scales	2020-07-27 17:00:00	8.5	15128.0	-1

Namun setelah data transaksi dihapus dan dicek kembali, ternyata masih ada juga *Quantity* dengan nilai negatif yang cukup besar pada dataset. Kebanyakan data tersebut merupakan data discount yang diberikan kepada *customer*. Tetapi yang aneh adalah ketika dilakukan pengecekan, ternyata *customer* tidak melakukan transaksi sama sekali pada hari dimana ia mendapatkan *discount*. Contoh datanya adalah sebagai berikut.

```
[55] customer_data[customer_data.Quantity<-1]
```

	Description	InvoiceDate	UnitPrice	CustomerID	Quantity
82431	discount	2020-05-07 11:00:00	0.01	16422.0	-720
82440	discount	2020-07-14 12:00:00	0.03	12901.0	-240
82441	discount	2020-07-14 14:00:00	1.45	16672.0	-18
82444	discount	2020-07-23 12:00:00	0.20	16029.0	-48
82449	discount	2020-08-18 14:00:00	0.70	16013.0	-96
253205	postage	2020-06-16 12:00:00	3.00	12463.0	-2
253964	postage	2021-01-27 10:00:00	1.00	12452.0	-2

Kemudian kita coba cek transaksi dari customer dengan ID 16422 dan tanggal transaksi 7 Mei 2020. Hasilnya adalah tidak ada transaksi sama sekali di hari itu (Mulai dari pukul 01.00 – 23.00) kecuali transaksi *discount*. Berdasarkan hal ini kita bisa putuskan untuk menghapus semua transaksi yang nilai *Quantity*nya lebih kecil dari -1.

```
#what he bought on that date  
customer_data[(customer_data.CustomerID==16422) & ((customer_data.InvoiceDate > '2020-05-07 01:00:00') & (customer_
```

	Description	InvoiceDate	UnitPrice	CustomerID	Quantity
82431	discount	2020-05-07 11:00:00	0.01	16422.0	-720

Setelah kita melauhi tahap *cleansing*, langkah terakhir yang perlu kita lakukan adalah mengecek nilai dari *summary statistic* dari data tersebut untuk memastikan data yang akan kita olah. Berikut ini nilai dari *summary statistic* terakhir sebelum data digunakan untuk membuat fitur RFM.

	UnitPrice	CustomerID	Quantity
count	386212.000000	386212.000000	386212.000000
mean	3.110961	15280.371793	12.944908
std	22.271629	1712.932604	48.266433
min	0.000000	12347.000000	-1.000000
25%	1.250000	13941.000000	2.000000
50%	1.950000	15144.000000	6.000000
75%	3.750000	16782.000000	12.000000
max	8142.750000	18287.000000	12540.000000

Nilai *UnitPrice* yang masih sangat tinggi itu bukan berasal dari harga produk, namun harga jasa pengiriman produk.

**Langkah ke 2** adalah membangun fitur RFM dimana caranya cukup sederhana yaitu dengan mengelompokkan data transaksi berdasarkan *CustomerID*. Berikut istilah yang perlu dipahami dalam membangun fitur RFM.

- *Recency (R)*: merupakan selisih jumlah hari antara tanggal terakhir *customer* melakukan transaksi dengan tanggal dilakukannya analisis. Misalkan *customer* terakhir melakukan transaksi pada tanggal 9 Februari 2021 dan analisis mulai dilakukan pada tanggal 12 Februari 2021, maka nilai *recency customer* tersebut adalah 3. Semakin kecil nilai *recency* maka akan semakin bagus, hal ini berarti *customer* merupakan orang baru atau orang yang sering berbelanja di *e-commerce* tersebut.
- *Frequency (F)*: berapa kali *customer* belanja di *e-commerce* tersebut, semakin sering *customer* berbelanja (nilai F semakin besar) maka bisa dikatakan *customer* nyaman dengan pelayanan yang diberikan. Cara menghitungnya adalah dengan menghitung jumlah unik Invoice number, atau jumlah data unik tanggal dan jam transaksi dari *customer*.
- *Monetary (M)*: berapa total uang yang dihabiskan oleh *customer* untuk belanja disana. Semakin banyak nilai dari *monetary*, berarti orang tersebut semakin loyal. Cara menghitungnya adalah menjumlahkan nilai  $Quantity \times UnitPrice$ .

Hasil dari proses ini bisa dilihat di bawah dengan jumlah *customer* sebanyak 4.337 orang.

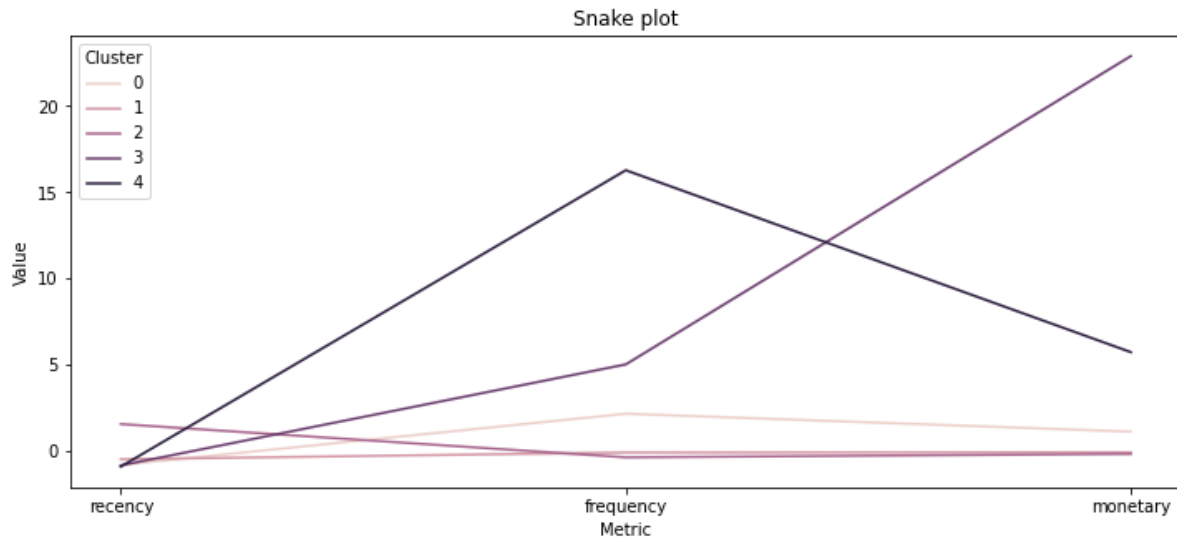
	CustomerID	recency	frequency	monetary
0	12347.0	2	7	4310.00
1	12348.0	75	4	1797.24
2	12349.0	19	1	1757.55
3	12350.0	310	1	334.40
4	12352.0	36	7	1545.41

**Langkah ke 3** adalah melakukan pengelompokan *customer* berdasarkan 3 fitur yang sudah dibuat (*recency*, *frequency* dan *monetary*). Proses ini bisanya dikenal dengan istilah *clustering* dimana salah satu algoritma yang bisa digunakan adalah *KMeans*. Berdasarkan proses *clustering* yang telah dilakukan, dihasilkan 5 kelompok data yang memiliki *characteristic* yang sama antar datanya. Adapun masing-masing karakternya bisa dilihat pada data di bawah ini.

	recency	frequency	monetary
Cluster			
0	11.38	18.37	11379.47
1	44.31	3.38	1306.56
2	248.88	1.50	518.17
3	9.00	37.40	195034.19
4	1.86	112.29	50302.14

Contoh cara membaca data tersebut:

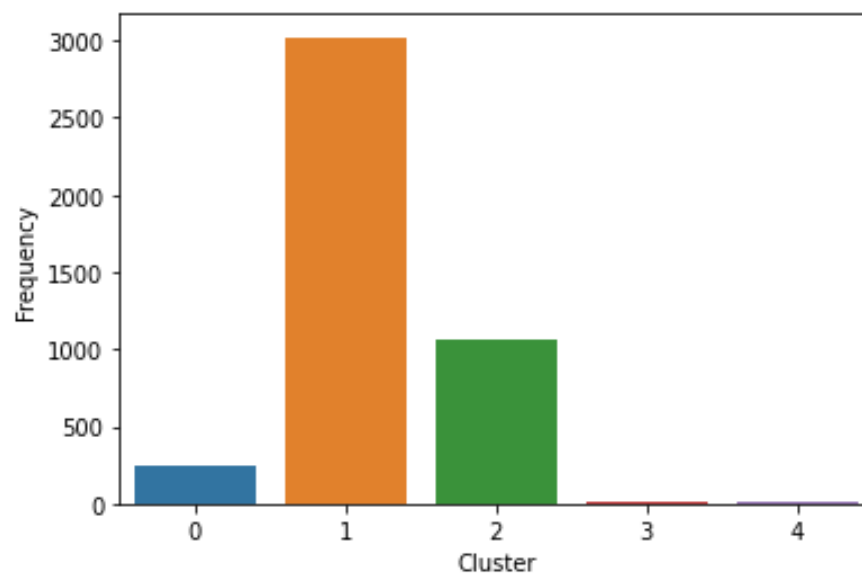
Cluster 4, berisikan *customer* yang sangat sering berbelanja dengan rata-rata 2 hari sekali mereka berbelanja (*recency*). Dilihat dari segi berapa kali mereka pernah belanja di *e-commerce* tersebut (*frequency*), nilainya pun sangat tinggi dimana rata-rata *customer* sudah belanja disana sebanyak 112 kali. Adapun untuk total belanjaan yang dihabiskan selama ini (*monetary*) rata-rata sebesar 50.302 dolar. *Customer* pada *cluster* ini bisa dikategorikan sebagai *loyal customer*. Untuk mempermudah memahaminya kita bisa melihat snake plot diagram untuk masing-masing *cluster* di bawah ini, sehingga bisa dengan mudah kita tahu perbedaan antar masing-masing *cluster*.



Penamaan *Cluster* (penamaan bisa disesuaikan dengan pengamatan *business analyst*):

- Cluster 0 : Normal Customer
- Cluster 1 : Risk Customer
- Cluster 2 : Sleeping Customer
- Cluster 3 : High Spender
- Cluster 4 : Loyal Customer

Sebaran dari masing-masing *cluster* bisa dilihat pada grafik di bawah ini. Adapun detail jumlah *customer* pada masing-masing cluster yaitu cluster 0: 240 orang, cluster 1: 3021 orang, cluster 2: 1064 orang, cluster 3: 5 orang dan cluster 4: 7 orang.



Berikut ini adalah hasil dari proses *clustering customer* yaitu data RFM masing-masing *customer* beserta *segment* dari *customer* itu sendiri. Dengan menggunakan data ini *manajer* akan tahu bahwa setiap *customer* masuk ke dalam cluster apa dan *treatment* apa yang cocok diberikan.

	CustomerID	recency	frequency	monetary	Cluster
0	12347.0	2	7	4310.00	1
1	12348.0	75	4	1797.24	1
2	12349.0	19	1	1757.55	1
3	12350.0	310	1	334.40	2
4	12352.0	36	7	1545.41	1

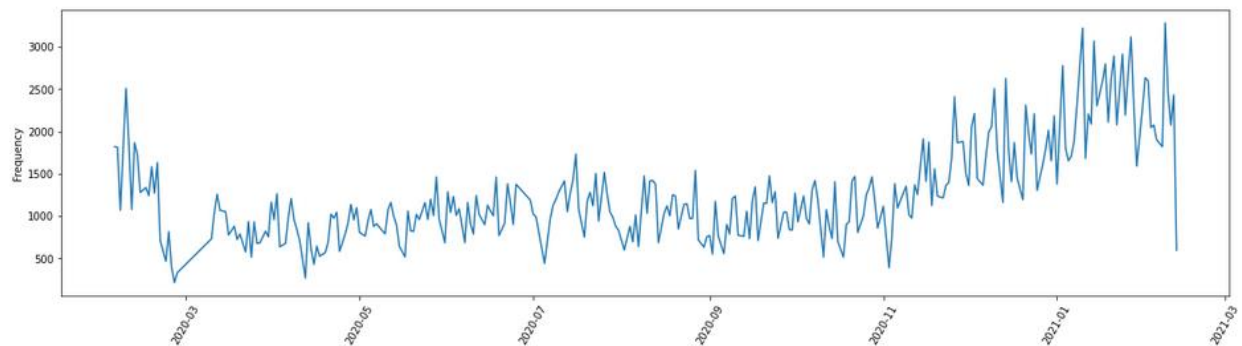
Berdasarkan data *clustering customer* tersebut, kita bisa tahu apa yang harus manajer tersebut lakukan terhadap setiap *segment* pelanggannya. **Berikut ini contoh insight yang mungkin bisa diterapkan:**

- Jaga relasi *customer* yang berada di *cluster* 3 dan 4 agar tetap berbelanja di *e-commerce* kita dengan cara sesering mungkin memberikan informasi produk yang sedang *trending* atau memberikan promo menarik kepada mereka. Misalkan untuk *customer* di *cluster* 4 diberikan promo pada produk-produk yang harganya mahal, sehingga nilai *monetary* dari *customer* tersebut bisa ditingkatkan. Untuk *customer* pada *cluster* 3, diberikan banyak info terhadap produk baru yang sedang *trending*, tidak perlu yang murah namun bisa juga diberi promo. Hal ini karena *cluster* 3 adalah orang yang *high spender* tapi frekuensinya kecil, harapannya agar mereka lebih sering berbelanja. Selain itu 2 kelompok ini bisa diberikan membership atau hadiah, agar mereka merasa dihargai dan semakin sering belanja di *e-commerce* tersebut.
- *Customer* di *cluster* 0 (normal), berikan promo produk yang umum dibeli (seperti produk kebutuhan sehari-hari) tidak perlu terlalu murah karena mereka setidaknya 2 minggu sekali berbelanja. Total uang yang dihabiskan pun tidak terlalu sedikit. Mungkin bisa sesekali diberikan bonus agar mereka semakin merasa nyaman untuk belanja disana.
- *Customer* di *cluster* 1 (*risk customer*), mereka adalah *customer* yang mendekati *sleeping customer* (sangat jarang berbelanja). Mungkin pemberian promosi seperti beli 2 gratis 1, promo pada produk yang murah, *free ongkir*, dll bisa meningkatkan antusias mereka untuk berbelanja lagi. Setidaknya kita coba cegah agar mereka tidak berhenti berbelanja.
- *Customer* 2 (*sleeping customer*), *customer* ini tidak perlu diberikan promo karena mereka hanya membeli ketika mereka butuh saja atau mereka hanya orang yang coba-coba di platform kita. Jika kita memberikan *campaign* kepada mereka, besar kemungkinan *campaign* tersebut juga gagal. Sehingga lebih baik tidak mengeluarkan biaya berlebih untuk mereka.

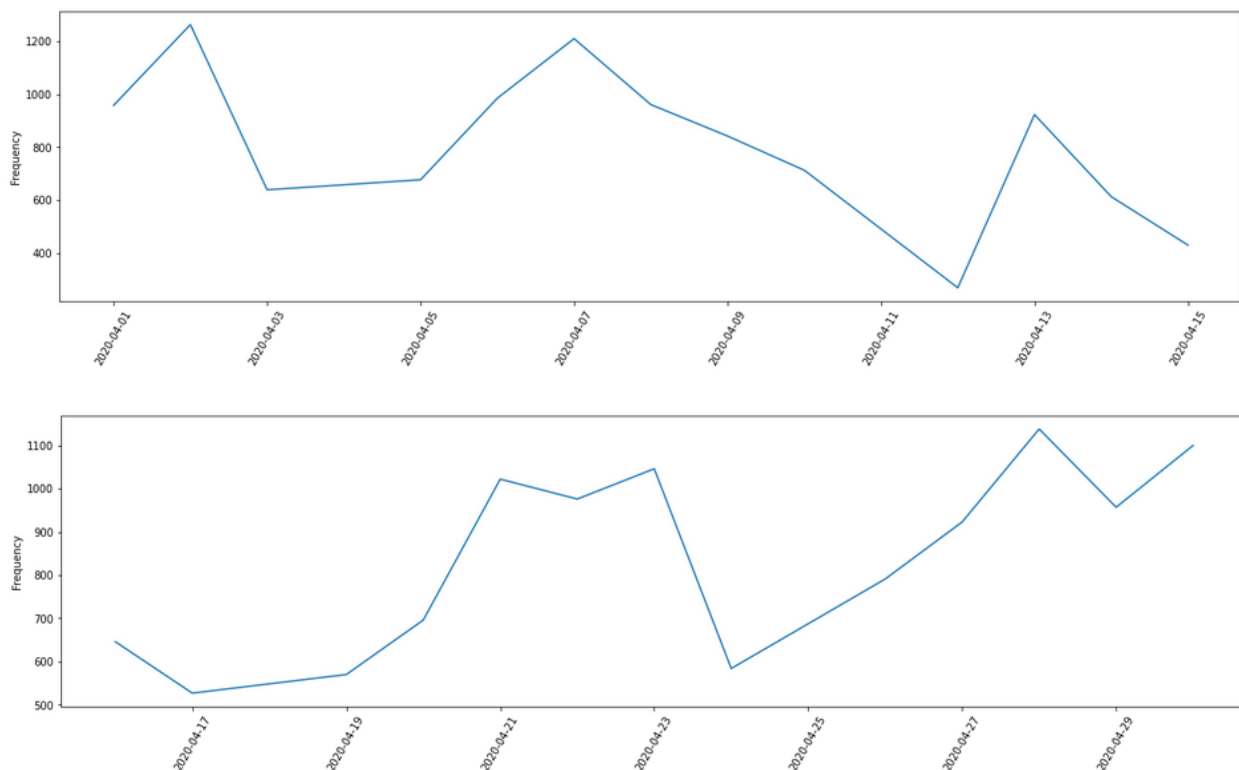
Setelah kita tahu kelompok *customer*, langkah selanjutnya adalah kapan waktu yang tepat untuk memberikan promo atau penawaran. Agar pada hari itu juga *income* yang masuk ke perusahaan tersebut semakin besar. Pada tahap ini kita bisa melihatnya dengan **time analysis**.

## TIME ANALYSIS

Untuk mengetahui kapan waktu yang tepat kapan suatu industri memberikan promo atau melakukan *launching* produk-produk baru, kita perlu tahu kapan terjadi transaksi dengan *traffic* yang tinggi. Hal ini bisa diketahui dengan cara menjumlah semua transaksi yang ada, kemudian dikelompokkan berdasarkan tanggal. Adapun hasil *traffic* transaksi untuk semua data dari awal transaksi hingga akhir transaksi bisa dilihat pada gambar di bawah ini.

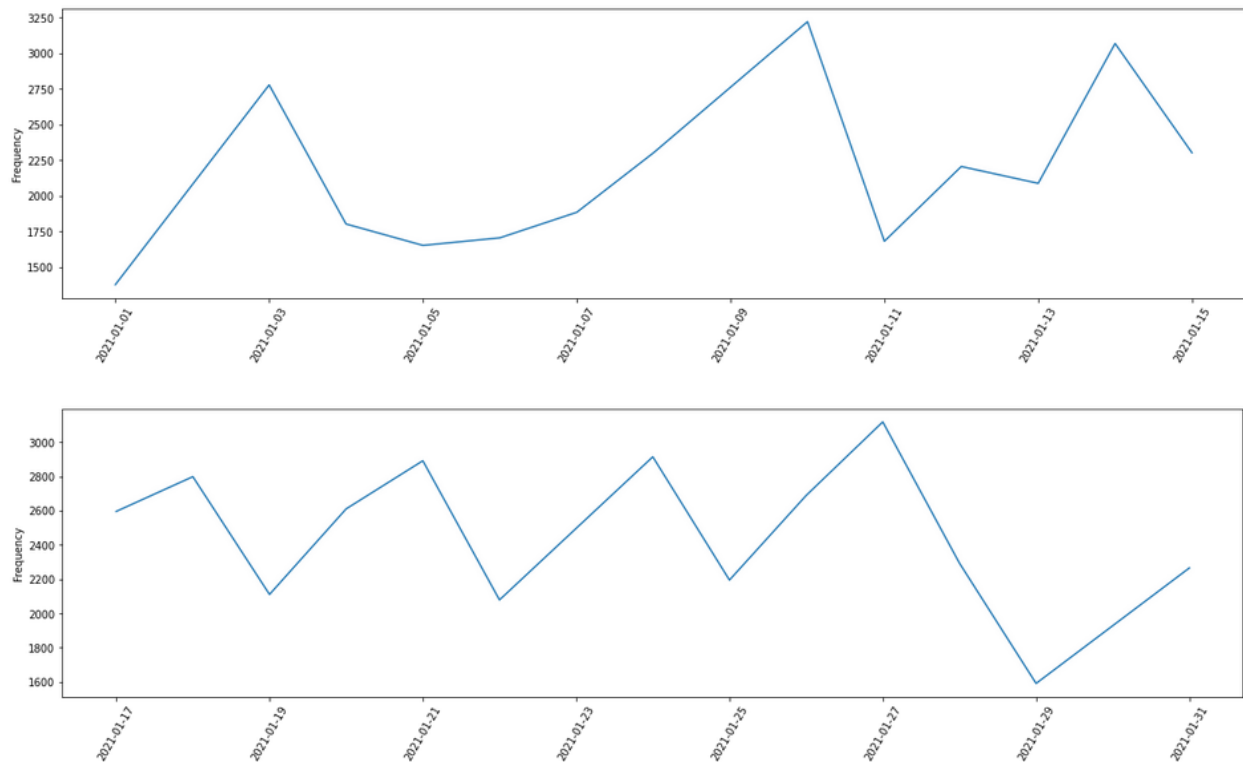


Karena data tersebut masih general, kita coba untuk menampilkan data yang lebih spesifik dengan hanya memilih bulan tertentu. Berikut grafik transaksi yang terjadi pada bulan April 2020.



Note: hari Sabtu berada pada tanggal 4, 11, 18 dan 25

Kemudian kita coba untuk ambil satu bulan lagi yaitu di bulan Januari 2021 dimana *traffic* nya mengalami peningkatan dari bulan sebelumnya.



Note: hari Sabtu berada pada tanggal 2, 9, 16, 23 dan 30

Perlu digaris bawahi bahwasannya sebelum diubah tahunnya ke tahun 2020-2021, transaksi paling awal terjadi pada tanggal 1 Desember 2010 dan paling akhir terjadi pada tanggal 9 Desember 2011. Sehingga grafik naik di bulan Januari 2021 sama dengan bulan November 2011.

Berikut ini **insight yang bisa diambil berdasarkan grafik total penjualan:**

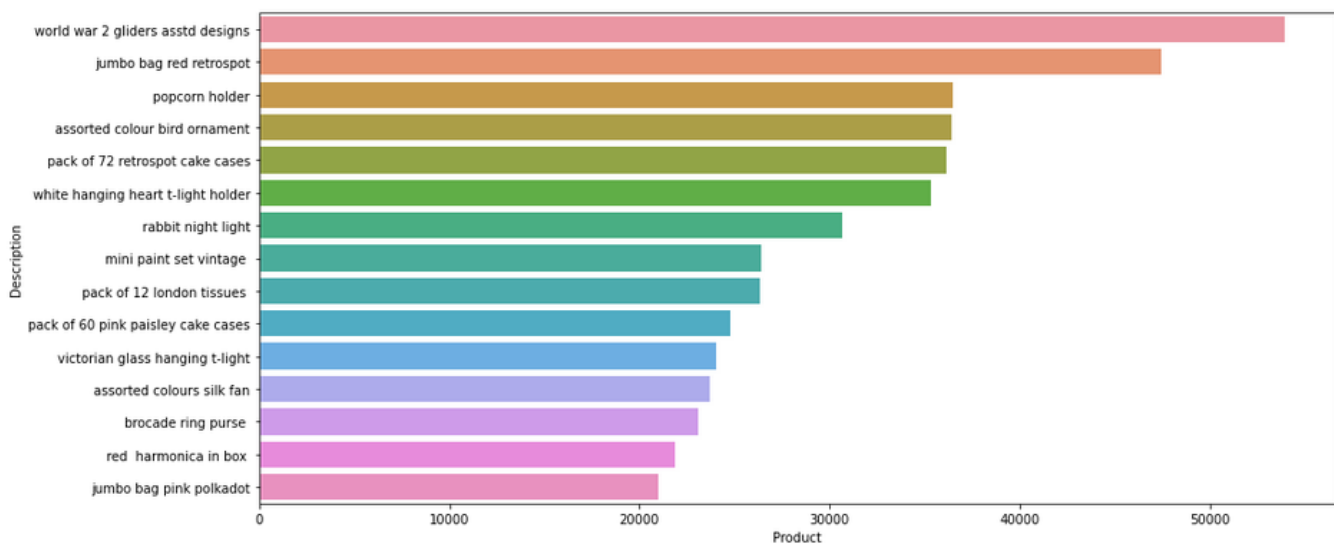
- Jika dilihat dari grafik keseluruhan transaksi, maka semakin mendekati akhir tahun maka banyak orang yang melakukan transaksi. Sehingga akhir tahun merupakan waktu yang tepat untuk *launching* produk baru, menawarkan promo akhir tahun atau memperbanyak *stock product*. Tujuannya cukup sederhana yaitu untuk mendapatkan untung sebanyak-banyaknya.
- Berdasarkan analisis transaksi data pada 2 bulan tersebut (April 2020 dan Januari 2021), rata-rata transaksi mengalami peningkatan di hari sabtu, minggu dan senin. Selain itu akhir bulan (sekitar tanggal 23) hingga awal bulan (sekitar tanggal 5) sering terjadi peningkatan transaksi juga. Kemungkinan besar ini terjadi karena banyak orang yang baru memperoleh gaji pada tanggal tersebut. Sehingga ini waktu yang tepat juga untuk melakukan mengeluarkan promo. Atau jika ingin menarik pelanggan di tanggal lain, kita bisa membuat promo untuk produk tertentu di pertengahan bulan. Tujuannya agar pada pertengahan bulan transaksi tidak terlalu turun.



Setelah kita tahu customernya siapa dan kapan promosi diberlakukan, langkah selanjutnya adalah produk apa yang akan dipromosikan. Untuk mengetahui hal tersebut, maka kita perlu melakukan *product analysis*.

## PRODUCT ANALYSIS

Produk analysis digunakan untuk mengetahui produk mana yang paling laku dan produk mana yang tidak diminati oleh *customer*. Untuk produk yang laku dipasaran, kita bisa memperbanyak stoknya agar selalu siap jika ada pembeli yang mencari. Sementara produk yang sudah jarang pembelinya bisa dikurangi produksinya atau bahkan dipromosikan untuk menghindari kerugian akibat produk rusak (terlalu lama disimpan). Adapun untuk produk *best-seller* bisa dilihat pada grafik di bawah ini.



Berikut ini detail jumlah dari produk-produk yang paling laris (kiri) dan produk yang sedikit peminatnya (kanan).

Description	Quantity	Description	Quantity
3965 world war 2 gliders asstd designs	53943	1209 etched glass star tree decoration	1
1837 jumbo bag red retrospot	47475	388 black glass/shell/pearl necklace	1
2705 popcorn holder	36459	2069 marie antioienett trinket box gold	1
234 assorted colour bird ornament	36389	3109 set 36 colouring pencils doiley	1
2364 pack of 72 retrospot cake cases	36132	3169 set of 3 pink flying ducks	1
3875 white hanging heart t-light holder	35320	479 blue new baroque flock candlestick	1
2765 rabbit night light	30691	1350 for online retail orders	1
2134 mini paint set vintage	26437	1109 dusty pink christmas tree 30cm	1
2330 pack of 12 london tissues	26315	3306 set/6 ivory bird t-light candles	1
2362 pack of 60 pink paisley cake cases	24757	2543 pink crystal guitar phone charm	1

Berikut ini **insight yang bisa diambil berdasarkan list produk terjual**:

- Produk yang sudah laku dalam jumlah besar tidak perlu dipromosikan lagi, namun lebih ke arah menjaga stok produk tersebut yang ada di gudang. Agar setiap pembeli yang mencari produk tersebut selalu mendapatkan apa yang ia cari.
- *Manajer* bisa coba untuk mempromosikan atau melakukan obral pada produk-produk yang sangat sedikit peminatnya. Hal ini dilakukan untuk cuci gudang dan meminimalisir angka kerugian akibat produk yang tidak laku dan akhirnya rusak. Kemudian jangan terlalu banyak memproduksi produk tersebut, atau jika memang samasekali tidak menguntungkan lebih baik untuk tidak menjual produk tersebut.
- Produk-produk yang berada ditengah-tengah (laku namun peminatnya tidak terlalu besar) mungkin bisa dipromosikan kembali untuk memperbesar peluang produk tersebut menjadi laku dan menambah banyak keuntungan.

## Kesimpulan

Untuk mengoptimalkan promosi pada *e-commerce*, setidaknya ada tiga hal yang perlu diperhatikan yaitu siapa target *customer* nya, kapan waktu yang tepat promo diberikan dan produk apa yang akan dipromosikan. Jika strategi analisis untuk ketiga hal ini berhasil maka keuntungan dari penjualan produk pun bisa dioptimalkan. Namun jika kita melakukan promosi tanpa strategi apapun, maka besar kemungkinan yang terjadi hanyalah promosi berjalan namun tidak ada yang berbeda dengan transaksi yang sudah berlangsung sebelumnya. Lebih mudahnya bisa dikatakan sebagai agenda bakar-bakar uang saja.

### Note:

Untuk full codenya bisa diakses disini [transaction analisis](#)