

# **VISUALISASI DATA ONLINE RETAIL TAHUN 2009-2011**



## **KELOMPOK MURID PROF**

<b>KENNETH HOLIVIAN TO</b>	<b>6162001018</b>
<b>JASON HINARDI</b>	<b>6162001022</b>
<b>BRYAN ERNESTIN</b>	<b>6162001097</b>
<b>LEONARDO ALINDRA</b>	<b>6162001111</b>

**UNIVERSITAS KATOLIK PARAHYANGAN**

**BANDUNG**

**2023**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Data merupakan informasi-informasi yang telah dikumpulkan. Namun, data sendiri belum dapat menghasilkan informasi baru jika tidak diproses dan dianalisis. Data yang sering ditemukan sekarang berukuran sangat besar sehingga diperlukan program dan bantuan komputer untuk memprosesnya. Dengan bantuan komputer, data dapat lebih mudah dicerna dan menghasilkan grafik atau visualisasi agar lebih mudah untuk mencapai suatu konklusi berdasarkan data yang digunakan.

Data yang akan digunakan adalah data “online\_retail\_II” yang berisi data penjualan suatu perusahaan retail di United Kingdom selama tahun 2009-2010 dan 2010-2011 yang dipisah ke dalam dua sheet. Informasi yang ada di dalam data tersebut diantaranya adalah invoice, kode stok, deskripsi produk, tanggal penjualan, jumlah yang dipesan, harga satuan, ID pembeli, dan negara asal pembeli. Namun, ada beberapa bagian dalam data yang tidak lengkap sehingga di baris tersebut masih memiliki cell kosong.

### **1.2 Rumusan Masalah**

Adapun rumusan masalah untuk karya tulis ini adalah sebagai berikut:

- Produk online retail apa yang paling laris selama tahun 2009-2011?
- Bagaimana segmentasi pelanggan online retail selama tahun 2009-2011?
- Bagaimana tren penjualan online retail selama tahun 2009-2011?
- Bagaimana perbandingan penjualan online retail dalam dan luar negeri selama tahun 2009-2011?
- Negara asing apa yang melakukan pembelian terbesar dari online retail selama tahun 2009-2011?
- Bagaimana tren penjualan online retail setiap jamnya selama tahun 2009-2011?

### **1.3 Tujuan Penelitian**

Adapun tujuan penelitian untuk karya tulis ini adalah sebagai berikut:

- Mengetahui produk online retail apa yang paling laris selama tahun 2009-2011.
- Mengetahui segmentasi pelanggan online retail selama tahun 2009-2011.
- Mengetahui tren penjualan online retail selama tahun 2009-2011.
- Mengetahui perbandingan penjualan online retail dalam dan luar negeri selama tahun 2009-2011.

- Mengetahui negara asing yang melakukan pembelian terbesar dari online retail selama tahun 2009-2011.
- Mengetahui tren penjualan online retail setiap jamnya selama tahun 2009-2011.

## BAB II

### ISI

#### 2.1 Proses ETL (Extract, Transform, Load)

Proses ini dilakukan pada Python. Langkah pertama adalah ekstrak data yang dibutuhkan. Lalu, dilanjutkan dengan memproses data sehingga menjadi lebih cocok untuk melakukan analisis. Setelah itu, data yang telah diproses diubah kembali ke dalam bentuk file yang diinginkan (file excel).

##### 2.1.1 Pembuatan Function

Pertama, *import library* Pandas dan Numpy. Setelah itu, dilanjutkan dengan pembuatan function-function untuk melakukan proses ETL.

1. Function `extract_excel` untuk ekstrak data dari file Excel

```
def extract_excel(file_path, sheet):  
    data = pd.read_excel(file_path, sheet_name = sheet)  
    return data
```

2. Function `change_col_name` untuk mengubah nama setiap kolom sesuai yang diinginkan

```
def change_col_name(data):  
    data = data.rename(columns={"Invoice" : "invoice", "StockCode" : "stock_code", "Description" : "description",  
                               "Quantity" : "quantity", "InvoiceDate" : "invoice_date", "Price" : "price",  
                               "Customer ID" : "customer_id", "Country" : "country"})  
    return data
```

Dalam menamakan suatu kolom, diinginkan nama tersebut tidak mengandung spasi dan huruf kapital.

3. Function `merge_data` untuk menggabungkan dua buah *dataframe* dengan *dataframe* pertama lalu dilanjutkan dengan *dataframe* kedua

```
def merge_data(data_a, data_b):  
    frames = [data_a, data_b]  
    merge = pd.concat(frames, ignore_index = True)  
    return merge
```

Parameter `ignore_index = True` digunakan untuk mengulang kembali index baris data sehingga index akan kembali 0, 1, ..., n-1 untuk suatu n baris data.

4. Function `convert_data_type` untuk mengubah tipe data dari suatu kolom menjadi yang diinginkan

```
def convert_data_type(data, column, data_type):  
    data[column] = data[column].astype(data_type)  
    return data
```

5. Function `remove_duplicate` untuk menghapus baris data yang duplikat

```
def remove_duplicate(data):  
    # Menghilangkan semua baris data yang sama kecuali baris data pertama  
    data = data.drop_duplicates(keep='first', ignore_index = True)  
    return data
```

Function ini akan menyimpan baris data yang muncul pertama dan menghapus semua baris data yang menjadi duplikatnya.

6. Function `remove_missing_value` untuk menghapus baris data yang memiliki informasi yang tidak lengkap atau kosong

```
def remove_missing_value(data):  
    data = data.dropna().reset_index(drop = True)  
    return data
```

Perintah `reset_index(drop = True)` serupa dengan `ignore_index = True` yang akan mengulang index dari baris data.

7. Function `remove_negative_qty` untuk menghapus baris data yang memiliki “quantity” negatif

```
def remove_negative_qty(data):  
    temp = []  
    for row in range(len(data.index)):  
        if data["quantity"][row]<0:  
            temp.append(row)  
    data_clean = data.drop(temp).reset_index(drop = True)  
    return data_clean
```

8. Function `total_order` untuk menghitung hasil perkalian dari “quantity” dan “price” untuk masing-masing baris data

```
def total_order(data):  
    total = []  
    for row in range(len(data.index)):  
        temp = data["quantity"][row] * data["price"][row]  
        total.append(temp)  
    data["total_order"] = total  
    return data
```

9. Function `produk_terlaris` untuk mencari produk terlaris berdasarkan suatu ukuran, yaitu “quantity” atau “total\_order”

```
def produk_terlaris(data, ukuran):  
    # Ukuran dapat berupa "quantity" atau "total_order"  
    code_group = data[["stock_code", ukuran]].groupby("stock_code").sum()  
    merge = pd.merge(code_group, df_produk, on="stock_code")  
    return merge.sort_values(ukuran, ascending=False).head()
```

10. Function pelanggan\_terbanyak untuk mengurutkan customer berdasarkan suatu ukuran, yaitu "quantity" atau "total\_order" dari yang terbesar

```
def pelanggan_terbanyak(data, ukuran):  
    # Ukuran dapat berupa "quantity" atau "total_order"  
    cust_group = data[["customer_id", ukuran]].groupby("customer_id").sum()  
    merge = pd.merge(cust_group, df_customer, on="customer_id")  
    return merge.sort_values(ukuran, ascending=False)
```

11. Function segmentasi\_pelanggan untuk melakukan klasifikasi atau segmentasi pelanggan berdasarkan kuartil dari ukuran "quantity" atau "total\_order"

```
def segmentasi_pelanggan(data, quartile, ukuran):  
    # Ukuran dapat berupa "quantity" atau "total_order"  
    segmentasi = []  
    for row in range(len(data.index)):  
        if data[ukuran][row] <= quartile[0]:  
            segmentasi.append("small")  
        elif data[ukuran][row] <= quartile[1]:  
            segmentasi.append("medium")  
        elif data[ukuran][row] <= quartile[2]:  
            segmentasi.append("high")  
        else:  
            segmentasi.append("very high")  
    data["cluster"] = segmentasi  
    return data
```

12. Function trend\_quarter untuk mencari produk terlaris pada suatu triwulan atau kuartal tahun

```
def trend_quarter(kuartal):  
    c = df_penjualan[df_penjualan["quarter"] == kuartal]  
    c1 = produk_terlaris(c, "quantity")  
    c2 = produk_terlaris(c, "total_order")  
    return c, c1, c2
```

13. Function order\_per\_jam untuk menghitung jumlah order pada setiap jam

```
def order_per_jam(data):  
    order_jam = data["hours"].value_counts()  
    return order_jam
```

14. Function `dalam_luar` untuk menghitung jumlah transaksi dalam dan luar negeri

```
def dalam_luar(data):  
    luar_negeri = 0  
    dalam_negeri = 0  
    for i in range(len(data)):  
        if data[i] == "United Kingdom":  
            dalam_negeri = dalam_negeri + 1  
        else:  
            luar_negeri = luar_negeri + 1  
    return luar_negeri, dalam_negeri
```

### 2.1.2 Proses Extract

Dalam proses extract, didefinisikan *file path* dimana *file* `online_retail_II` disimpan dalam komputer. Lalu dilanjutkan dengan mengekstrak masing-masing *sheet* dari *file* tersebut dan disimpan ke dalam variabel `data_1` dan `data_2`. Proses ini ditunjukkan oleh gambar di bawah.

```
path = r"C:/Users/ASUS/OneDrive/Documents/Keperluan UNPAR/Semester 6/Kapita Selektat Statistika/UTS/online_retail_II.xlsx"  
data_1 = extract_excel(path, "Year 2009-2010")  
data_2 = extract_excel(path, "Year 2010-2011")
```

### 2.1.3 Proses Transform

Langkah awal pada proses transform adalah mengganti nama kolom dari masing-masing `data_1` dan `data_2`. Lalu, mengubah tipe data pada kolom “invoice”, “stock\_code”, “description”, “customer\_id”, dan “country” menjadi tipe *string* karena data pada kolom ini tidak akan digunakan untuk perhitungan matematis.

```
data_1 = change_col_name(data_1)  
data_2 = change_col_name(data_2)  
  
joint_data = merge_data(data_1, data_2)  
  
joint_data = convert_data_type(joint_data, ["invoice", "stock_code", "description", "customer_id", "country"], "string")
```

Selanjutnya, terdapat data yang terduplikat pada `data_2` yang merupakan data penjualan tahun 2010-2011. Baris-baris data duplikat tersebut kemudian dibuang agar tidak terjadi kekeliruan dalam analisis data nantinya. Dalam data yang diberikan, juga terdapat baris-baris data yang memiliki informasi tidak lengkap dan/atau “quantity” negatif sehingga baris-baris data tersebut juga dihapus.

```
joint_data = remove_duplicate(joint_data)  
  
data_nonan = remove_missing_value(joint_data)  
  
data_nonneg = remove_negative_qty(data_nonan)
```

Setelah itu, data pada kolom “quantity” diubah ke dalam bentuk bilangan bulat atau *integer*. Lalu, `data_full` diperoleh dengan menambahkan kolom `total_order` sebagai hasil

perkalian dari “quantity” dan “price” masing-masing baris data.

```
data_noneg = convert_data_type(data_noneg, "quantity", "int")
data_full = total_order(data_noneg)
```

Perhatikan bahwa `data_full` adalah *dataframe* dengan informasi setiap kolom secara lengkap. *Dataframe* tersebut dapat dipecah menjadi 3 buah *dataframe* yang lebih sederhana, yaitu *dataframe* yang memuat informasi tentang customer (`df_customer`), produk (`df_produk`), dan , penjualan (`df_penjualan`). *Dataframe* `df_customer` memuat informasi tentang "customer\_id" dan "country". *Dataframe* `df_produk` memuat informasi tentang "stock\_code" dan "description". *Dataframe* `df_penjualan` memuat informasi tentang "invoice", "invoice\_date", "customer\_id", "stock\_code", "quantity", "price", dan "total\_order".

```
df_customer = remove_duplicate(data_full[["customer_id", "country"]])
df_produk = remove_duplicate(data_full[["stock_code", "description"]])
df_penjualan = data_full[["invoice", "invoice_date", "customer_id", "stock_code", "quantity", "price", "total_order"]]
df_penjualan_awal = df_penjualan
```

Analisis data yang pertama adalah mencari produk terlaris berdasarkan “quantity” dan “total\_order”. Lalu, hasilnya dicetak untuk melihat lima produk terlaris berdasarkan kedua jenis ukuran.

```
# Produk Terlaris berdasarkan quantity
prod_laris_qty = produk_terlaris(df_penjualan, "quantity")
print(prod_laris_qty)

# Produk Terlaris berdasarkan total_order
prod_laris_total = produk_terlaris(df_penjualan, "total_order")
print(prod_laris_total)
```

Analisis data yang kedua adalah melakukan segmentasi pelanggan menjadi empat buah cluster, yaitu “small”, “medium”, “high”, dan “very high” berdasarkan “quantity” dan “total\_order”. Pertama, pelanggan diurutkan dari yang terbesar berdasarkan masing-masing ukuran. Selanjutnya, dicari kuartil 1, 2, dan 3 dari data tersebut. Pelanggan dengan jumlah ukuran yang lebih kecil dari kuartil 1 akan masuk ke cluster “small”, yang lebih kecil dari kuartil 2 akan masuk ke cluster “medium”, dan seterusnya hingga masing-masing pelanggan masuk ke dalam satu dari empat cluster.

```
# Segmentasi Pelanggan berdasarkan total_order
segmen_1 = pelanggan_terbanyak(df_penjualan, "total_order").reset_index(drop = True)
quartile_1 = numpy.quantile(segmen_1["total_order"], [0.25,0.5,0.75])
segmen_total_order = segmentasi_pelanggan(segmen_1, quartile_1, "total_order")

# Segmentasi Pelanggan berdasarkan quantity
segmen_2 = pelanggan_terbanyak(df_penjualan, "quantity").reset_index(drop = True)
quartile_2 = numpy.quantile(segmen_2["quantity"], [0.25,0.5,0.75])
segmen_quantity = segmentasi_pelanggan(segmen_2, quartile_2, "quantity")
```



Analisis data yang ketiga adalah mencari produk terlaris pada setiap triwulan atau kuartal tahun. Pertama, masing-masing baris data dikelompokkan ke dalam satu dari empat kuartal (*quarter*). Dalam hal ini, tahun transaksi tidak berpengaruh terhadap pengelompokkan. Selanjutnya, dicari produk terlaris pada setiap kuartal tahun berdasarkan ukuran “quantity” dan “total\_order” dan kemudian hasilnya dicetak.

```
# Produk terlaris (per Kuartal Tahun)
df_penjualan["quarter"] = df_penjualan["invoice_date"].dt.quarter

# Kuartal 1
quarter1 = trend_quarter(1)[1:3] # Produk terlaris pada kuartal 1 (quantity dan total_order)
print(quarter1)
c1 = trend_quarter(1)[0]
c1_final = pd.merge(c1, df_produk, on="stock_code")

# Kuartal 2
quarter2 = trend_quarter(2)[1:3] # Produk terlaris pada kuartal 2 (quantity dan total_order)
print(quarter2)
c2 = trend_quarter(2)[0]
c2_final = pd.merge(c2, df_produk, on="stock_code")

# Kuartal 3
quarter3 = trend_quarter(3)[1:3] # Produk terlaris pada kuartal 3 (quantity dan total_order)
print(quarter3)
c3 = trend_quarter(3)[0]
c3_final = pd.merge(c3, df_produk, on="stock_code")

# Kuartal 4
quarter4 = trend_quarter(4)[1:3] # Produk terlaris pada kuartal 4 (quantity dan total_order)
print(quarter4)
c4 = trend_quarter(4)[0]
c4_final = pd.merge(c4, df_produk, on="stock_code")
```

Analisis data yang keempat adalah mencari jumlah transaksi yang terjadi pada setiap jam. Setiap baris data dikelompokkan ke dalam jam terjadinya transaksi. Lalu, dihitung jumlah transaksi yang terjadi pada masing-masing jam.

```
# Jumlah Order per Jam
df_penjualan["hours"] = df_penjualan["invoice_date"].dt.hour

banyak_order = order_per_jam(df_penjualan)
print(banyak_order)
df_banyak_order = pd.DataFrame({
    'time_in_hours': ['12', '13', '14', '11', '15',
                     '10', '16', '9', '17', '8',
                     '19', '18', '20', '7', '6'],
    'orders_count': [138995, 126957, 106717, 96858, 86151,
                    72308, 51109, 40984, 26246, 15183,
                    7897, 7179, 1857, 1053, 1]
})
```

Analisis data yang terakhir adalah mencari jumlah penjualan dalam dan luar negeri. Dalam data ini, suatu transaksi termasuk penjualan dalam negeri jika pembeli berasal dari United

Kingdom dan suatu transaksi termasuk penjualan luar negeri jika pembeli tidak berasal dari United Kingdom. Lalu, dihitung jumlah penjualan dalam dan luar negeri.

```
# Penjualan Dalam dan Luar Negeri
cust_penjualan = pd.merge(df_penjualan, df_customer, on = "customer_id")
country = cust_penjualan["country"]

banyak_transaksi = dalam_luar(country)
print(banyak_transaksi)
df_banyak_transaksi = pd.DataFrame({
    'transaksi': ['luar_negeri', 'dalam_negeri'],
    'banyak': [banyak_transaksi[0], banyak_transaksi[1]]
})
```

#### 2.1.4 Proses Load

Proses ini dilakukan dengan menconvert *dataframe* yang dihasilkan dari proses transform ke dalam bentuk *file* excel kembali. Proses ini ditunjukkan oleh gambar di bawah.

```
data_full.to_excel("df_full_data_awal.xlsx")

df_customer.to_excel("df_customer.xlsx")
df_produk.to_excel("df_produk.xlsx")
df_penjualan_awal.to_excel("df_penjualan_awal.xlsx")

segmen_total_order.to_excel("segmentasi_total_order.xlsx")
segmen_quantity.to_excel("segmentasi_quantity.xlsx")

c1_final.to_excel("quarter_1.xlsx")
c2_final.to_excel("quarter_2.xlsx")
c3_final.to_excel("quarter_3.xlsx")
c4_final.to_excel("quarter_4.xlsx")

df_banyak_order.to_excel("banyak_order_tiap_jam.xlsx")

df_banyak_transaksi.to_excel("banyak_transaksi_dalam_luar_negeri.xlsx")

penjualan_produk = pd.merge(df_penjualan, df_produk, on="stock_code")
data_full_final = pd.merge(penjualan_produk, df_customer, on="customer_id")

data_full_final.to_excel("df_full_data_final.xlsx")
```

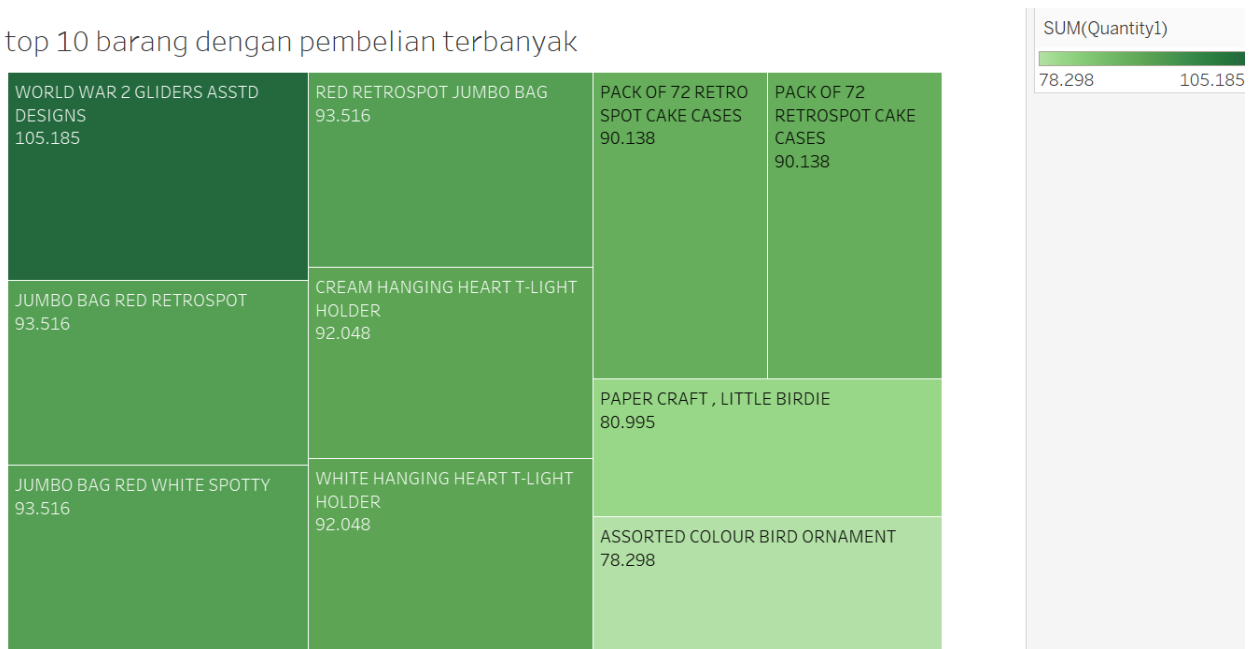
Setelah itu, *file* excel yang telah dibersihkan dimasukkan ke dalam Tableau untuk melakukan visualisasi data.

## 2.2 Visualisasi Data

### 2.2.1 Produk Terlaris

Produk terlaris dapat dilihat berdasarkan seberapa banyak suatu produk dibeli atau berdasarkan total penjualan dari suatu produk. Untuk melihat barang yang sering dibeli, kolom informasi yang akan digunakan adalah kolom “description” dan “quantity”, lalu semua baris data yang memiliki deskripsi yang sama akan dijumlahkan kolom kuantitas data tersebut. Dengan cara ini, didapatkan hasil visualisasi sebagai berikut:

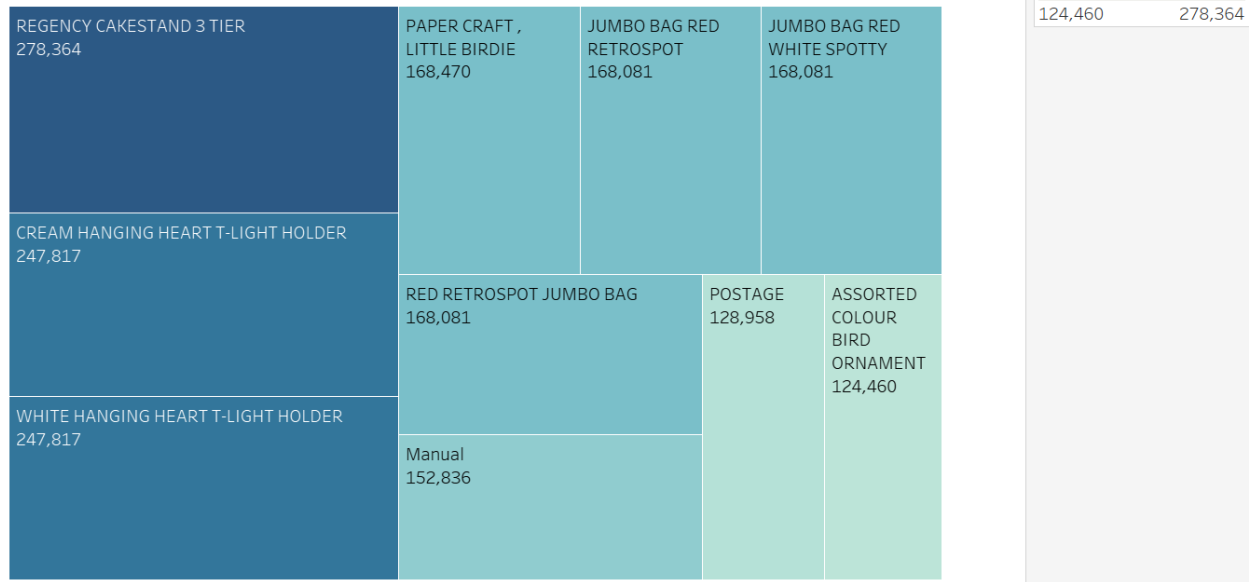
top 10 barang dengan pembelian terbanyak



Bisa dilihat bahwa produk “World War 2 gliders ASSTD Designs” merupakan produk yang paling laris dengan jumlah pembelian diatas 100,000. Setelah itu, dilanjutkan dengan produk tas, T-light holder, dan cake case yang masing-masing memiliki lebih dari satu variasi dan masing-masing terjual sebanyak 90,000. Karena produk-produk ini diminati, perusahaan online retail harus menyimpan stok yang lebih lebih banyak untuk produk-produk ini dan lebih berhati-hati supaya tidak kehabisan stok ketika permintaannya sudah banyak.

Sedangkan, jika dilihat dari besar pendapatannya berdasarkan kolom “total order”, didapatkan visualisasi sebagai berikut:

top 10 barang dengan penghasilan terbanyak

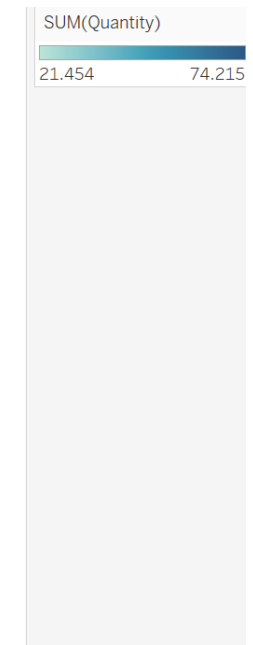
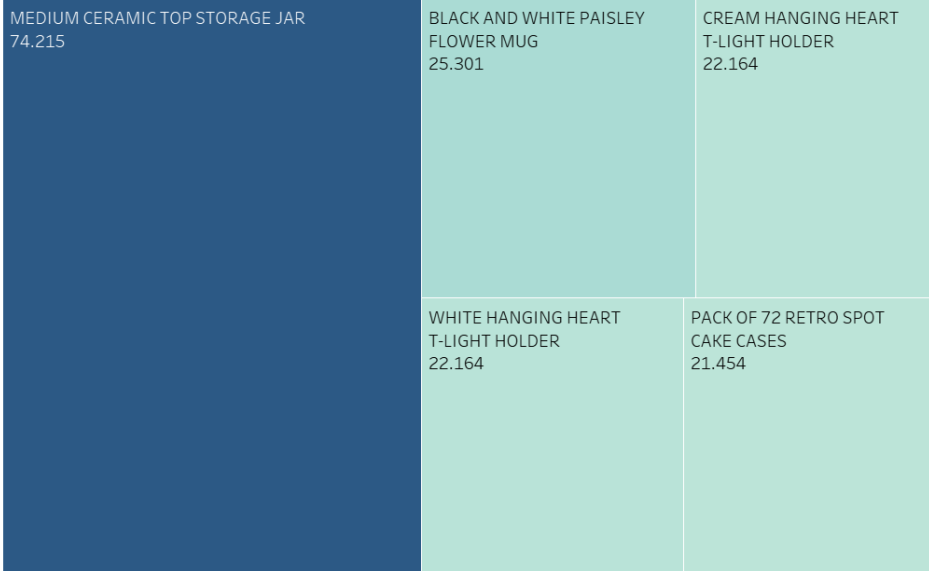


Bisa dilihat bahwa produk yang menghasilkan pendapatan terbesar adalah “Regency Cakestand 3 Tier”, dilanjutkan dengan T-light holder, paper craft, lalu produk tas. Jika dibandingkan dengan grafik sebelumnya, produk “World War 2 gliders ASSTD Designs” tidak menghasilkan pendapatan yang cukup besar meskipun memiliki penjualan yang banyak, diduga karena harga satuannya tidak terlalu besar. Sebaliknya, “Regency Cakestand 3 Tier” menghasilkan pendapatan yang besar meski tidak berada di sepuluh penjualan terbesar berdasarkan kuantitas, bisa diduga karena harga produk tersebut besar. Kemiripan yang bisa dilihat adalah T-light holder dan produk tas beserta variasinya memiliki jumlah penjualan dan pendapatan yang besar. Jadi, kedua produk ini sangat signifikan dalam kontribusi terhadap penjualan dan pendapatan.

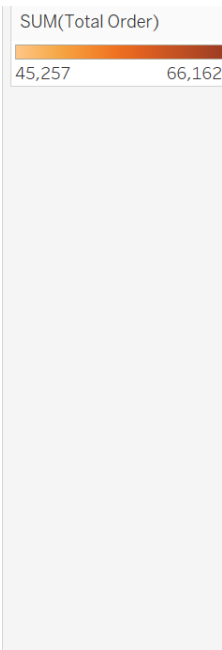
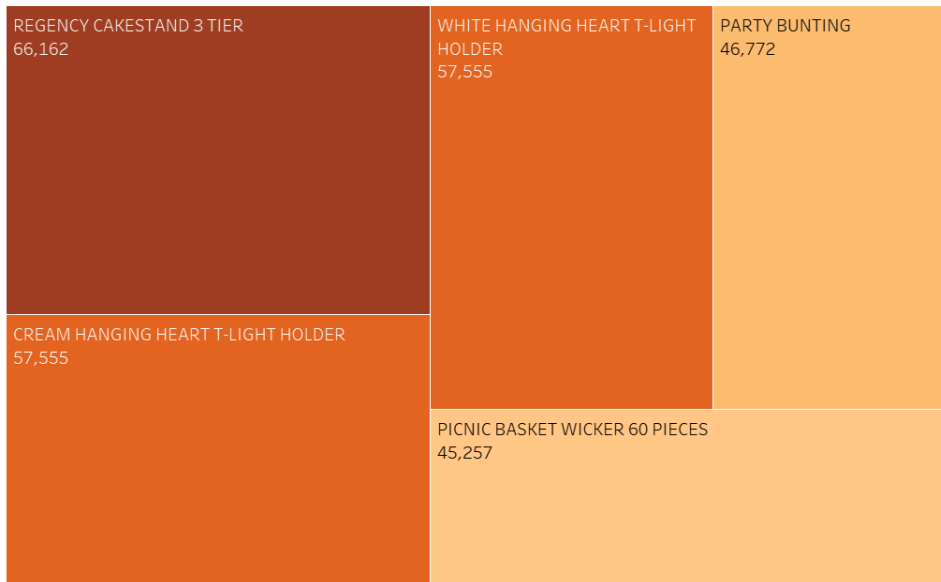
## 2.2.2 Produk Terlaris Setiap Triwulan

Untuk menganalisis lebih lanjut, peneliti mencari tahu total penjualan di setiap triwulan selama periode 2009-2011. Data akan diproses menggunakan Python agar satu file excel hanya menampilkan pembelian selama satu triwulan, lalu buat file excel baru untuk setiap triwulan lainnya. Dengan metode ini didapatkan visualisasi produk pembelian terbanyak dan keuntungan terbanyak di setiap triwulan sebagai berikut:

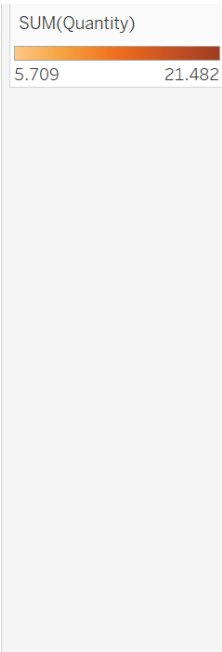
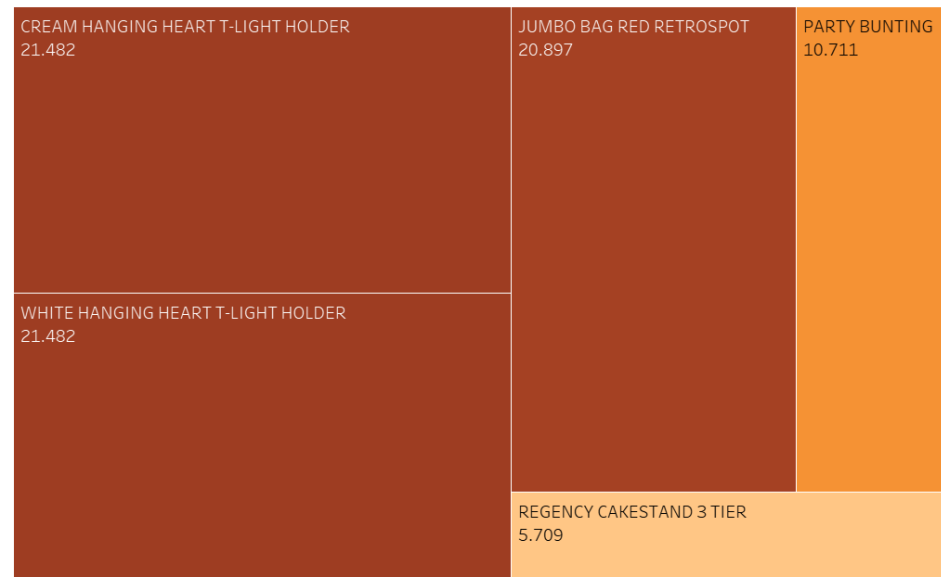
Product Category	Count
MEDIUM CERAMIC TOP STORAGE JAR	77,184
WHITE HANGING HEART T-LIGHT HOLDER	59,659
CREAM HANGING HEART T-LIGHT HOLDER	59,659
Manual	52,021
REGENCY CAKESTAND 3 TIER	43,067
MEDIUM CERAMIC TOP STORAGE JAR	43,067



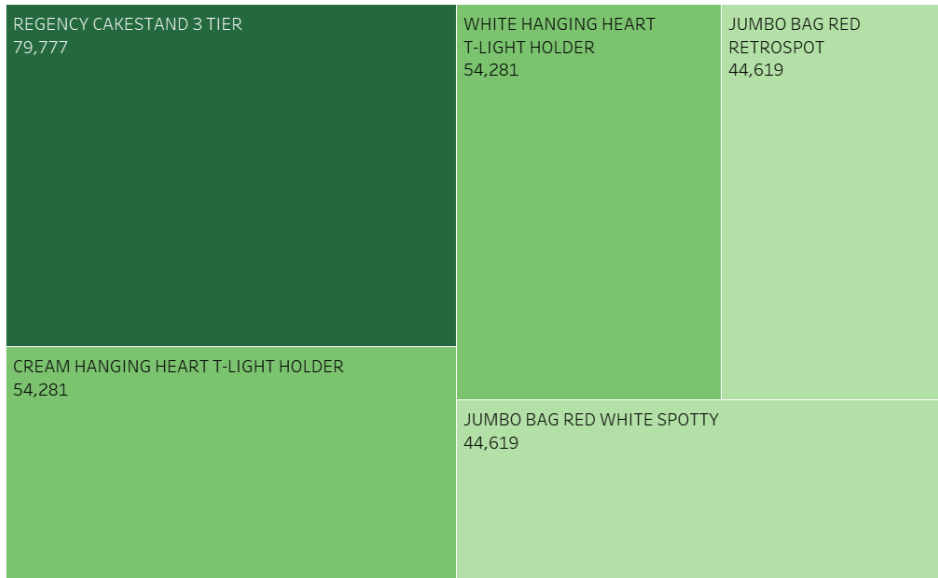
Top 5 Produk Pembelian Terbanyak pada Triwulan 2



Top 5 Produk Penghasilan Terbanyak pada Triwulan 2

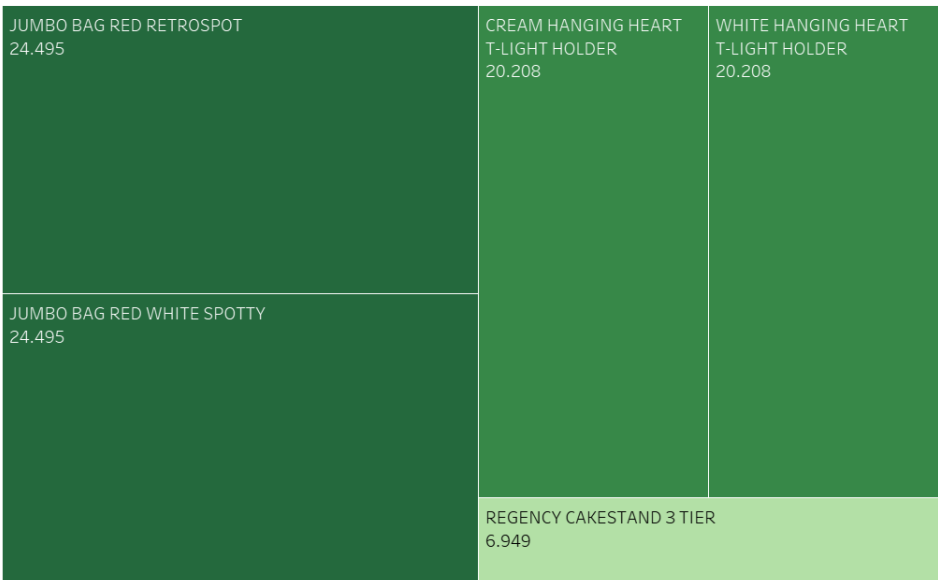


Top 5 Produk Pembelian Terbanyak pada Triwulan 3



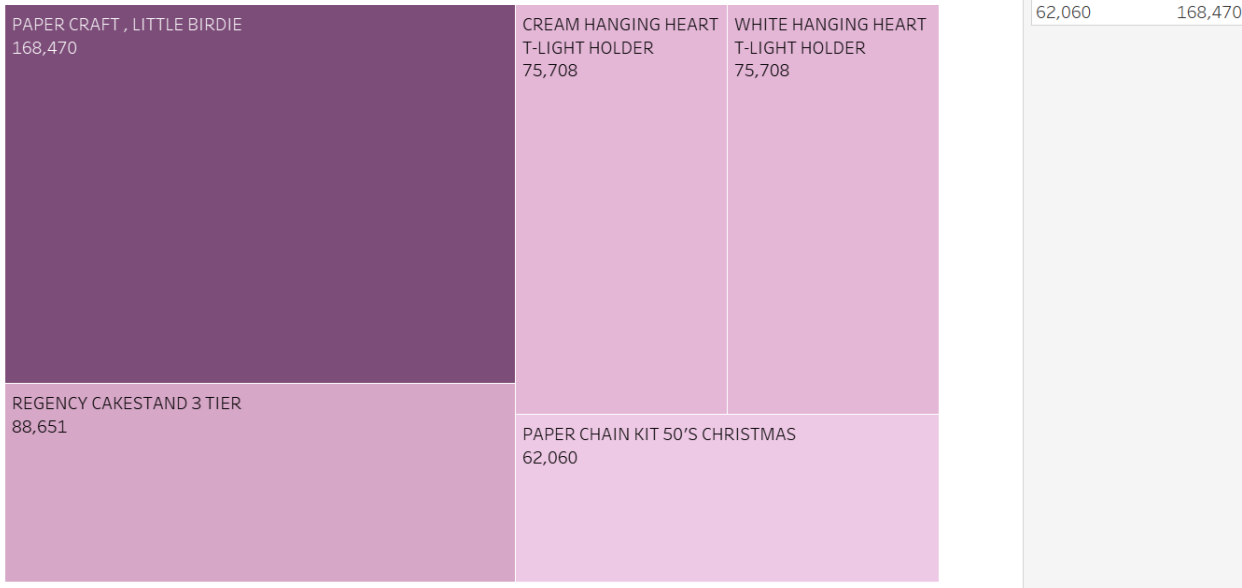
SUM(Total Order)	
44,619	79,777

Top 5 Produk Penghasilan Terbanyak pada Triwulan 3

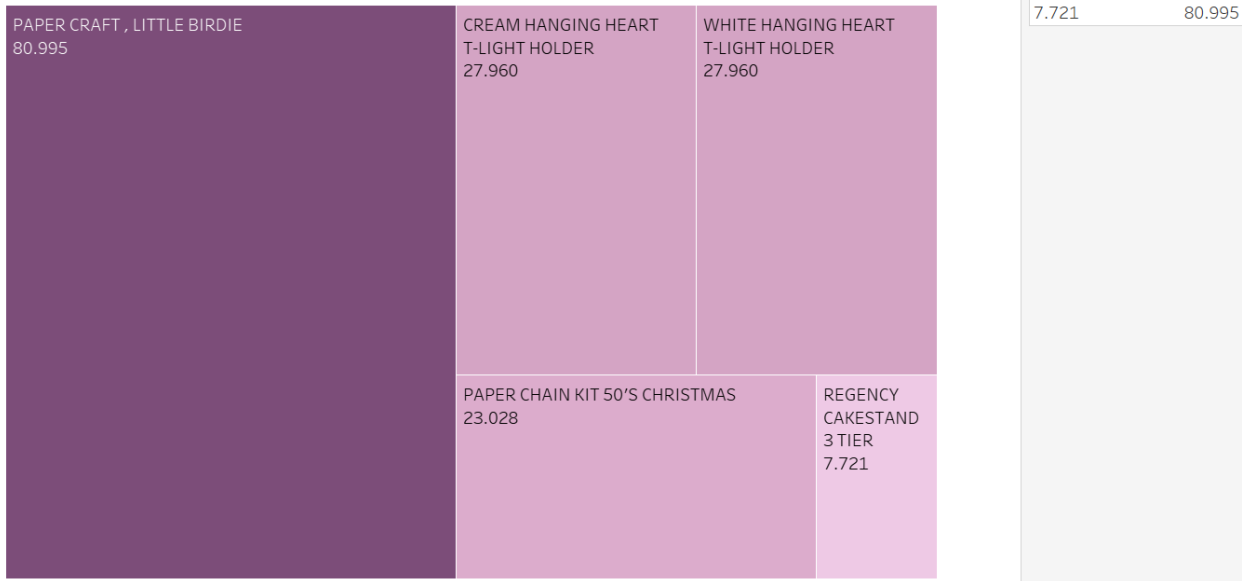


SUM(Quantity)	
6,949	24,495

Top 5 Produk Pembelian Terbanyak pada Triwulan 4



Top 5 Produk PenjualanTerbanyak pada Triwulan 4



Bisa dilihat bahwa produk yang paling banyak terjual di setiap triwulan adalah “Medium ceramic storage jar” pada triwulan pertama, “Regency cakestand 3 tier” dan “Cream Hanging Heart T-light Holder” pada triwulan kedua, “Regency cakestand 3 tier” dan “Jumbo Bag Red Retrospot” pada triwulan ketiga, dan “Paper Craft, Little Birdie” pada triwulan keempat. Oleh karena itu, toko dapat memperbanyak *stock* produk terlaris sebelum dimulainya setiap triwulan yang bersesuaian. Ketika semua visualisasi ini dianalisis secara keseluruhan, bisa dilihat bahwa produk “Regency Cakestand 3 Tier” dan “Jumbo Bag Red Retrospot” merupakan produk yang terlaris pada lebih dari satu triwulan. Tidak hanya itu, “Cream Hanging Heart T-light Holder” merupakan produk yang laris terjual dan memberikan pendapatan yang besar di setiap triwulan.

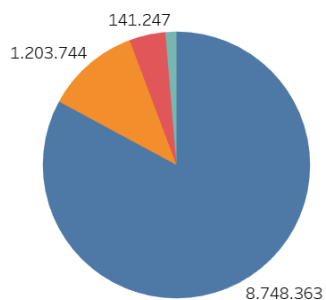


Maka itu, *stock* ketiga produk ini harus terus dijaga agar tidak kehabisan, terutama T-light holder yang laris sepanjang tahun.

## 2.3 Segmentasi Pelanggan

Segmentasi pelanggan dilakukan dengan cara menjumlahkan total pembelian setiap pelanggan, lalu dibagi ke dalam empat kategori berdasarkan jumlah pembelian dengan banyaknya pelanggan di setiap kategori yang sama. Dengan metode ini, didapatkan hasil visualisasi sebagai berikut:

Segmentasi Pelanggan berdasarkan Kuantitas

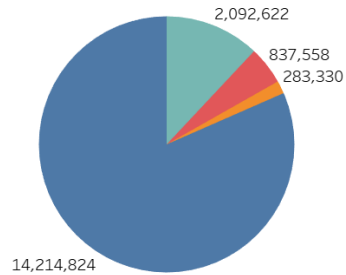


Cluster
<div><div></div>very high</div>
<div><div></div>high</div>
<div><div></div>medium</div>
<div><div></div>small</div>

SUM(Quantity)
10.554.828

Bisa dilihat bahwa pembelian terbesar sangat terkonsentrasi pada pelanggan yang membeli dengan kuantitas yang sangat banyak, bahkan melebihi tujuh kali lipat jumlah pembelian kategori berikutnya. Jika dilihat berdasarkan total pendapatan, didapatkan segmentasi sebagai berikut:

segmentasi pelanggan berdasarkan jumlah pesanan



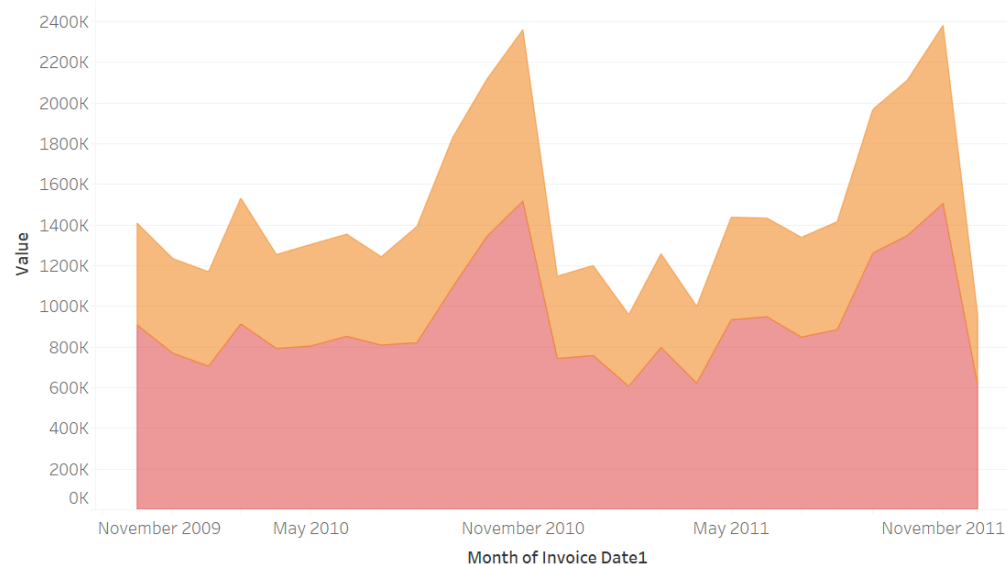
Cluster
high
medium
small
very high
SUM(Total Order)
17,428,335

Sama seperti kesimpulan visualisasi sebelumnya, total pendapatan sangat terkonsentrasi pada pembelian yang sangat banyak. Bisa dikatakan bahwa mayoritas pendapatan didapatkan dari pelanggan yang membeli secara masif. Maka itu, Perusahaan online retail harus mengutamakan pelanggan-pelanggan yang membeli secara masif untuk terus membeli produk dari perusahaan online retail ini.

## 2.4 Tren Penjualan

Untuk melihat secara garis besar kapan penjualan paling banyak terjadi, bisa dilihat di tren penjualan. Untuk itu, data akan ditampilkan dalam bentuk tren yang menunjukkan jumlah penjualan di tiap bulannya dan melihat perubahannya. Didapatkan visualisasi tren dari tahun 2009-2011 sebagai berikut:

tren penjualan musiman 2009-2011



Measure Names

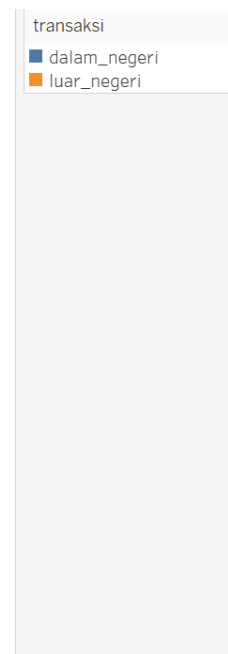
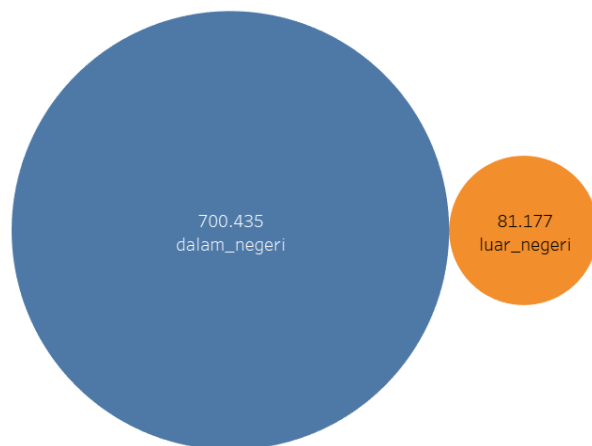
Quantity1  
Total Order1

Bisa dilihat pada bulan November di kedua tahun, terjadi peningkatan penjualan yang signifikan dibandingkan pada bulan lainnya. Dengan informasi ini, ada kemungkinan besar juga bahwa pada bulan November yang akan datang, akan terjadi peningkatan penjualan yang drastis juga.

## 2.5 Perbandingan Penjualan Dalam Negeri dan Luar Negeri

Dengan melihat perbandingan penjualan dalam negeri dan luar negeri, bisa dilihat apakah perusahaan online retail ini sangat fokus hanya ke dalam negeri atau sudah banyak ekspor barang. Perusahaan online retail ini berpusat di United Kingdom, jadi penjualan di dalam United Kingdom tersebut dianggap penjualan dalam negeri, sedangkan penjualan ke negara selain United Kingdom dianggap penjualan ke luar negeri. Berikut adalah hasil visualisasi

Banyaknya Transaksi Dalam Negeri dan Luar Negeri



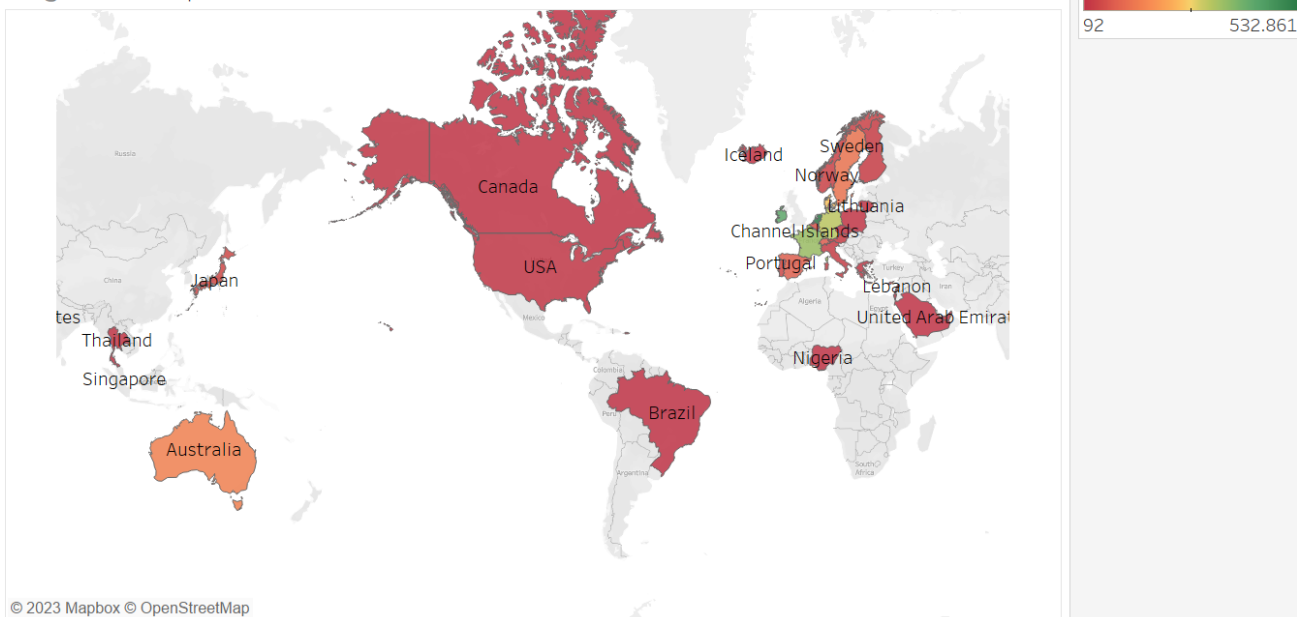
perbandingan kuantitas produk yang terjual dalam negeri dan luar negeri:

Bisa dilihat bahwa jumlah produk yang dijual ke dalam negeri sekitar 90% dan ekspor hanya sekitar 10%. Penjualan ke luar negeri bisa dianggap masih sedikit, salah satu kemungkinannya karena ongkos kirim ke luar negeri yang ditanggung pembeli lebih mahal daripada di dalam negeri. Jika permasalahan ini dan beberapa kendala dalam ekspor barang dapat ditangani, perusahaan online retail ini masih memiliki potensi yang besar jika ingin melakukan ekspor dengan lebih banyak.

## 2.6 Negara Asing dengan Pembelian Terbanyak

Untuk melihat negara asing mana yang paling banyak membeli produk, maka kita membuat visualisasi berupa peta dunia yang dibedakan berdasarkan warna. Warna tersebut merepresentasikan kuantitas produk yang dibeli oleh setiap negara. Dengan mengetahui laporan tersebut, maka kita dapat berfokus pada negara yang melakukan banyak pembelian.

Kegiatan Ekspor Berdasarkan Kuantitas

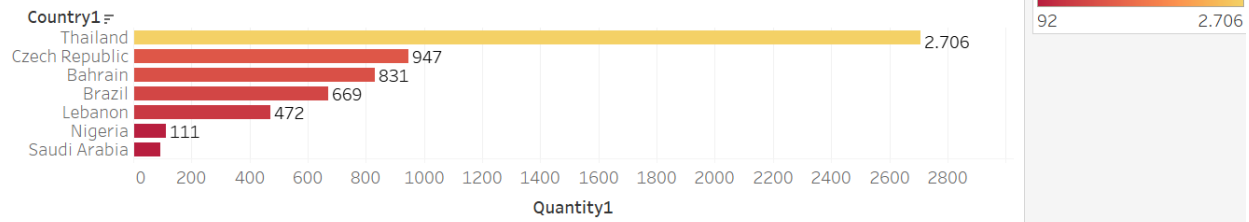


Dari visualisasi tersebut, didapat bahwa negara asing yang paling banyak melakukan pembelian adalah Belanda, EIRE, Prancis, Jerman, dan Denmark. Jika perusahaan online retail ingin meningkatkan ekspor produk, perusahaan tersebut dapat mengutamakan negara-negara yang memang sudah memiliki banyak pembeli terlebih dahulu sebelum mengarah ke negara lain. Dari sini dapat dilihat bahwa negara-negara yang melakukan pembelian terbanyak berada di sekitar United Kingdom.

## 2.7 Negara dengan Pembelian Terkecil

Dari hasil pemrosesan data, didapat bahwa negara dengan pembelian terkecil adalah Saudi Arabia dengan jumlah pembelian sebanyak 92 barang. Agar lebih jelas, dibuat visualisasi sebagai berikut:

Top 7 Negara dengan Pembelian Terkecil

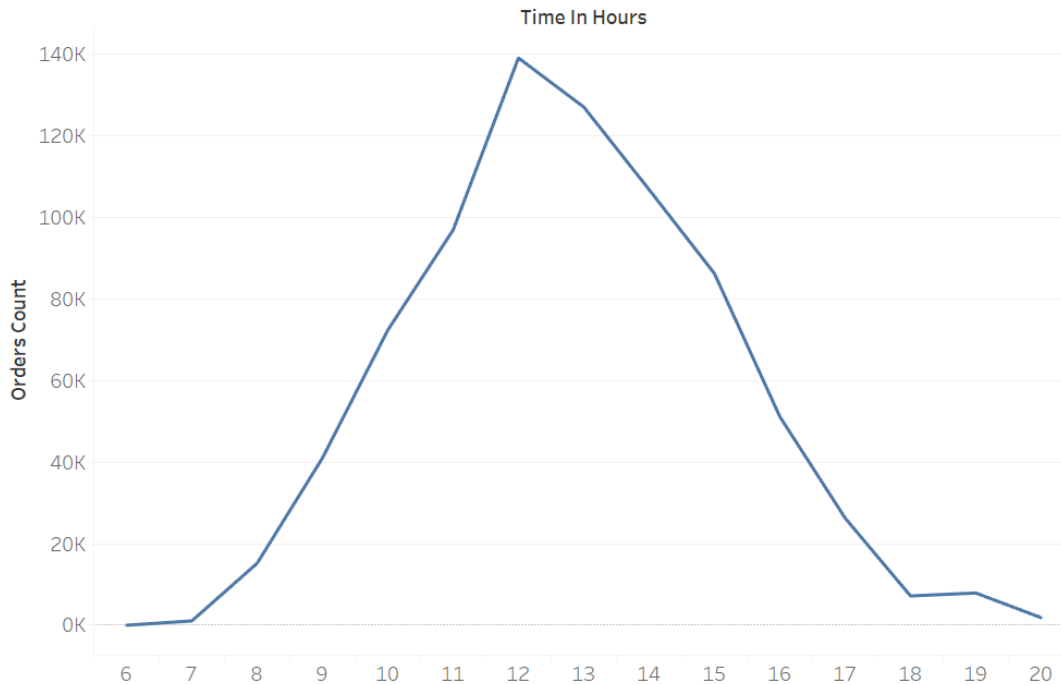


Dari 7 negara tersebut, 6 negara berada cukup jauh dari United Kingdom. Maka dari itu, kami berasumsi hal ini terjadi karena ongkos pengiriman yang cukup mahal. Kami menyarankan untuk memberikan bebas ongkir bagi 7 negara tersebut untuk mendorong keinginan konsumen di negara tersebut agar melakukan pembelian di perusahaan ini. Hal ini berlaku jika konsumen memenuhi syarat minimum kuantitas atau besar pembelian agar perusahaan tidak mengalami kerugian.

## 2.8 Penjualan per Jam

Dengan melihat penjualan per jam, bisa diketahui kapan server online retail harus diperbesar dan menghindari “overcrowding” pada server. Informasi yang digunakan adalah jam transaksi dan banyak pembelian. Peneliti menggunakan Python untuk membuat suatu file excel yang hanya mengambil jam transaksi dan menjumlahkan banyaknya kuantitas yang terjadi pada setiap jamnya, kemudian divisualisasikan. Dengan metode ini, didapatkan visualisasi sebagai berikut:

banyaknya order tiap jam



Bisa dilihat bahwa pemesanan terbanyak terjadi pada sekitar jam 12, dengan total 140,000 pembelian pada jam tersebut selama periode 2009-2011. Bisa dilihat juga bahwa perubahan jumlah pembelian sekitar jam 12 sangat berbeda dengan jumlah pembelian di jam lain, sehingga perusahaan online retail ini harus memastikan bahwa server tidak terjadi “overload” pada jam tersebut.

## **BAB III**

### **KESIMPULAN DAN SARAN**

#### **3.1 Kesimpulan**

Dengan menggunakan bantuan Python dan Tableau, data dapat diproses dan divisualisasikan untuk menghasilkan suatu konklusi. Dalam penelitian ini, visualisasi data terhadap data perusahaan online retail tahun 2009-2011 bisa menghasilkan informasi baru yang dapat menjadi panduan untuk meningkatkan penjualan dan pengembangan perusahaan.

Dari data yang diproses, didapat bahwa produk “World War 2 Gliders ASSTD Designs” merupakan produk dengan pembelian terbanyak, sedangkan produk “Regency Cakestand 3 Tier” merupakan produk yang menghasilkan pendapatan terbesar untuk toko tersebut.

Dilihat dari segmentasinya, jumlah pembelian dan total pendapatan sangat terkonsentrasi pada pembeli pada cluster very high.

Penjualan dalam negeri menghasilkan pendapatan perusahaan yang jauh lebih besar dibandingkan penjualan luar negeri.

Berdasarkan tren penjualan, mayoritas transaksi terjadi sekitar jam 12 siang dan puncak transaksi terjadi pada bulan November.

#### **3.2 Saran**

Adapun saran yang dapat diberikan berdasarkan kesimpulan di atas untuk meningkatkan penjualan dan perkembangan perusahaan adalah:

- Meningkatkan kapasitas server sekitar jam 12 siang dan pada bulan November agar tidak terjadi “overcrowd”.
- Mempertimbangkan untuk meningkatkan penjualan ke luar negeri, sebab penjualan ke luar negeri sementara masih sangat sedikit sehingga masih banyak peluang bisnis yang terbuka lebar. Hal ini dapat dilakukan dengan memperbanyak promosi ke luar negeri.
- Memberlakukan bebas ongkir untuk negara-negara dengan pembelian terendah dengan minimum kuantitas pembelian dengan tujuan untuk meningkatkan pembelian oleh negara-negara tersebut.