

## **Customers Report**

Lukito Andriansyah

# **Project in DQ Lab: Customers Report**

Project Mandiri Analisis Data

DQ Lab

lukitoandriansyah45@gmail.com

### Ringkasan

Betambahnya pelanggan tiap waktu serta meningkatnya pendapatan dalam suatu bisnis adalah harapan utama bagi pengemban bisnis itu sendiri. Di sisi lain, melihat performa tersebut juga menjadi salah satu indikator pihak manajemen dalam mengambil suatu kebijakan terkait bisnis itu sendiri. Proyek ini dilakukan untuk menelusuri performa transaksi yang terjadi di salah satu *retail* selama 2 kuartal pertama pada 2004. Data diperoleh dari DQ Lab yang kemudian dilakukan sedikit modifikasi dalam Microsoft Excel 2007. Data kemudian disimpan dan diolah dalam *data warehouse* di MySQL Workbench 8.0. Hasil menunjukkan bahwa terjadi penurunan performa yang signifikan di quarter ke-2 dengan nilai penurunan penjualan dan *revenue* masing-masing adalah 22% dan 24%. Selain itu, *retention rate* juga rendah, yakni 24% sehingga diperlukan strategi baru dalam melakukan banyak improvisasi baik dari segi produk ataupun marketing bisnis demi keberlangsungan bisnis kedepannya.

*Kata kunci:* Marketing Bisnis, Penjualan, *Retention Rate*, *Revenue*

## Pendahuluan

Melihat performa transaksi termasuk penjualan dan pendapatan serta melihat performa pelanggan yang datang merupakan salah satu cara dalam suatu bisnis untuk membantu dalam pengambilan keputusan baik dalam hal pengembangan produk bisnis itu sendiri ataupun pengembangan kinerja secara internal (Andry, 2018).

Proyek ini dilakukan untuk menelusuri performa transaksi yang terjadi di salah satu *retail* selama 2 kuartal pertama pada 2004. Data diperoleh dari DQ Lab yang kemudian dilakukan sedikit modifikasi dalam Microsoft Excel 2007. Data kemudian disimpan dan diolah dalam *data warehouse* di MySQL Workbench 8.0.

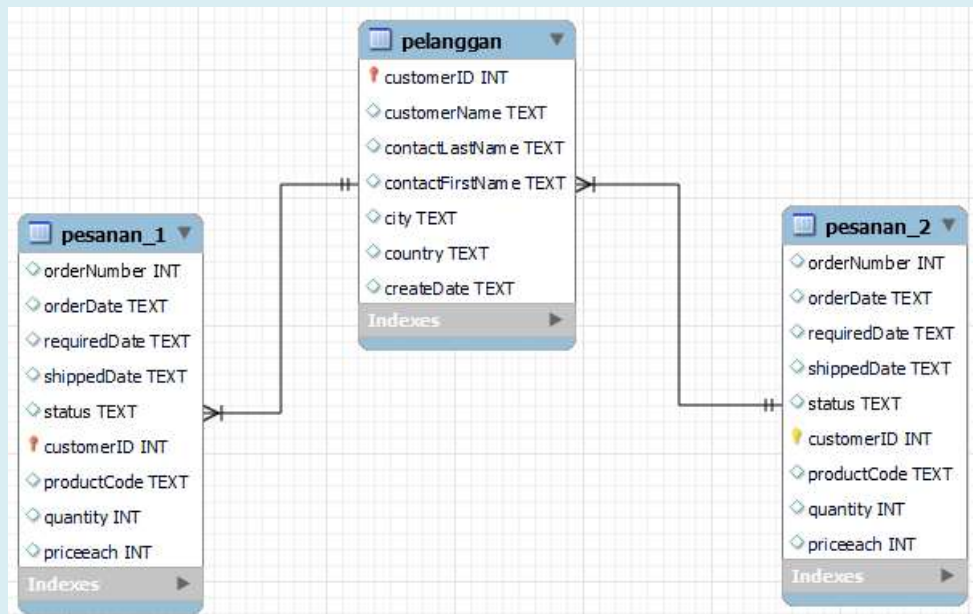
## Metode

### Alat

Proyek ini dilakukan secara mandiri dengan memanfaatkan Laptop, Microsoft Excel 2007, dan MySQL Workbench 8.0.

### Data

Data terdiri dari 3 data set, yakni *pesanan\_1*, *pelanggan*, dan *pesanan\_2*. Setiap dataset tentu memiliki kaitan satu sama lain seperti yang ditunjuk pada gambar berikut ini (DQ Lab, 2018):



Gambar 1 Relasi tiap-dataset

Perlu diketahui bahwa data tersebut belum dalam bentuk file. Sehingga kita perlu melakukan beberapa trik agar data tersebut bisa diolah di MySQL Workbench 8.0.

### Persiapan Data

Oleh karena kita memerlukan data tersebut untuk diolah dalam MySQL Workbench 8.0, maka kita harus meng-*import* data tersebut ke dalam MySQL

Workbench 8.0. Masalahnya adalah, data yang kita miliki bukanlah dalam bentuk file dan kita pun tidak memiliki akses langsung ke dalam data tersebut untuk bisa melakukan *import* data. Oleh karena itu, kita perlu melakukan beberapa trik sebagai berikut:

1. Salin semua data dari web tersebut ke dalam microsoft excel 2007.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2		orderNumber	orderDate	requiredDate	shippedDate	status	customerID	productCode	quantity	priceeach									
3																			
4		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S18_2581	24	81950									
5		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_1785	23	89720									
6		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_3949	33	55270									
7		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_4278	40	63030									
8		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_1374	41	90900									
9		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_4289	34	66730									
10		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S50_1341	41	37090									
11		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_1691	25	88600									
12		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2466	38	92740									
13		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2834	25	116200									
14		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_3167	32	73600									
15		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_4002	34	70330									

2. Hapus Rows yang berisi selain tanda “|”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2		orderNumber	orderDate	requiredDate	shippedDate	status	customerID	productCode	quantity	priceeach									
3																			
4		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S18_2581	24	81950									
5		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_1785	23	89720									
6		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_3949	33	55270									
7		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_4278	40	63030									
8		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_1374	41	90900									
9		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_4289	34	66730									
10		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S50_1341	41	37090									
11		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_1691	25	88600									
12		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2466	38	92740									
13		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2834	25	116200									
14		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_3167	32	73600									
15		10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_4002	34	70330									
16		10236	2004-04-03	2004-04-11	2004-04-08	Shipped	486	S10_2016	22	105860									
17		10236	2004-04-03	2004-04-11	2004-04-08	Shipped	486	S18_2625	23	52700									

3. Modifikasi data tersebut menggunakan fungsi find() dan replace() untuk mengganti semua tanda “|” menjadi semicolon “;”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		orderNumber;orderDate;requiredDate;shippedDate;status;customerID;productCode;quantity;priceeach;																	
2		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S18_2581;24;81950;																	
3		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S24_1785;23;89720;																	
4		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S24_3949;33;55270;																	
5		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S24_4278;40;63030;																	
6		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S32_1374;41;90900;																	
7		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S32_4289;34;66730;																	
8		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S50_1341;41;37090;																	
9		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S700_1691;25;88600;																	
10		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S700_2466;38;92740;																	
11		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S700_2834;25;116200;																	
12		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S700_3167;32;73600;																	
13		10235;2004-04-02;2004-04-12;2004-04-06;Shipped;260;S700_4002;34;70330;																	
14		10236;2004-04-03;2004-04-11;2004-04-08;Shipped;486;S10_2016;22;105860;																	
15		10236;2004-04-03;2004-04-11;2004-04-08;Shipped;486;S18_2625;23;52700;																	

Selesai, selanjutnya kita lakukan *import* data ke dalam MySQL Workbench 8.0

## Pengolahan Data

Terdapat tiga proses yang akan kita lalui, yaitu memeriksa dataset yang telah dilakukan import, memeriksa pertumbuhan penjualan, dan melakukan analisis pelanggan.



### 1. Memastikan DataSet yang Telah Dibuat

Pengecekan data disini hanya untuk mengetahui jenis data yang ada untuk dilakukan analisis. Gunakan query berikut ini:

```
SELECT * FROM `(project)_customer_report`.pelanggan
LIMIT 10;
SELECT * FROM `(project)_customer_report`.pesanan_1
LIMIT 10;
SELECT * FROM `(project)_customer_report`. pesanan_2
LIMIT 10;
```

Jika dijalankan, masing-masing keluaran yang dihasilkan adalah seperti yang ada pada gambart berikut ini:

	customerID	customerName	contactLastName	contactFirstName	city	country	createDate
▶	103	Atelier graphique	Schmitt	Carine	Nantes	France	2004-02-05
	112	Signal Gift Stores	King	Jean	Las Vegas	USA	2004-02-05
	114	Australian Collectors Co.	Ferguson	Peter	Melbourne	Australia	2004-02-20
	119	La Rochelle Gifts	Labrun	Janine	Nantes	France	2004-02-05
	121	Baane Mini Imports	Bergulfsen	Jonas	Stavern	Norway	2004-02-05
	124	Mini Gifts Distributors Ltd.	Nelson	Susan	San Rafael	USA	2004-03-11
	125	Havel & Zbyszek Co	Piestrzeniewicz	Zbyszek	Warszawa	Poland	2004-02-05
	128	Blauer See Auto Co.	Keitel	Roland	Frankfurt	Germany	2004-03-15
	129	Mini Wheels Co.	Murphy	Julie	San Francisco	USA	2004-02-05
	131	Land of Toys Inc.	Lee	Kwai	NYC	USA	2004-05-07

Gambar 2 Output query 10 data tabel pelanggan

	orderNumber	orderDate	requiredDate	shippedDate	status	customerID	productCode	quantity	priceeach
▶	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S72_1253	40	45690
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S700_2047	29	83280
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S24_3816	31	78830
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S24_3420	25	65090
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S24_2841	44	67140
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S18_3856	39	85750
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S18_3029	48	84300
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S18_1662	50	146650
	10234	2004-03-30	2004-04-05	2004-04-02	Shipped	412	S10_4757	48	118320
	10233	2004-03-29	2004-04-04	2004-04-02	Shipped	328	S700_2610	29	67940

Gambar 3 Output query 10 data tabel pesanan\_1

	orderNumber	orderDate	requiredDate	shippedDate	status	customerID	productCode	quantity	priceeach
▶	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S18_2581	24	81950
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_1785	23	89720
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_3949	33	55270
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S24_4278	40	63030
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_1374	41	90900
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S32_4289	34	66730
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S50_1341	41	37090
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_1691	25	88600
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2466	38	92740
	10235	2004-04-02	2004-04-12	2004-04-06	Shipped	260	S700_2834	25	116280

Gambar 4 Output query 10 data tabel pesanan\_2

Perlu diketahui bahwa, yang membedakan antara pesanan\_1 dengan pesanan\_2 adalah waktu transaksinya. Dengan kata lain, pesanan\_1 hanya mencatat transaksi yang terjadi dalam kuartal 1, sementara pesanan\_2 hanya mencatat transaksi yang terjadi dalam kuartal 2.

## 2. Periksa Pertumbuhan Penjualan

Selanjutnya yang harus kita lakukan adalah melakukan pemeriksaan terhadap pertumbuhan penjualan. Terdapat tiga sub-pertanyaan yang akan dipecahkan pada poin ini, yaitu:

### a. Total Penjualan dan Revenue

Sub-pertanyaan ini dilakukan pada tabel pesanan\_1 dan pesanan\_2. Total revenue merupakan perhitungan paling sederhana untuk mengetahui pendapatan yang diperoleh dalam suatu transaksi (Ibnu, 2020). Formulasi umumnya adalah seperti berikut ini:

$$TR = Q \times P$$

Dengan TR adalah total revenue, Q adalah jumlah barang yang diproduksi/terjual dan P adalah harga barang tersebut per item. Salah satu query yang bisa dilakukan adalah seperti berikut ini:

```
SELECT SUM(quantity) Total_Penjualan, SUM(quantity*priceeach)
Total_Revenue FROM `(project)_customer_report`.pesanan_1
WHERE status = 'Shipped';
SELECT SUM(quantity) Total_Penjualan, SUM(quantity*priceeach)
Total_Revenue FROM `(project)_customer_report`.pesanan_2
WHERE status = 'Shipped';
```

Keluaran dari query tersebut masing-masing adalah sebagai berikut:

	Total_Penjualan	Total_Revenue
▶	8694	799579310

Gambar 5 Output query Total\_Penjualan dan Total\_Revenue tabel pesanan\_1

	Total_Penjualan	Total_Revenue
▶	6717	607548320

Gambar 6 Output query Total\_Penjualan dan Total\_Revenue tabel pesanan\_2

### b. Presentasi Keseluruhan Penjualan

Presentasi keseluruhan, berarti kita perlu menyatukan dahulu kedua tabel pesanan\_1 dengan pesanan\_2. Kita bisa memanfaatkan fungsi **UNION** karena struktur tabel nanti akan digabung secara vertikal. Setelah itu lakukan query untuk menampilkan Total\_Penjualan dan Total Revenue dari penggabungan tabel tersebut beserta quater nya. Salah satu Query yang bisa dilakukan adalah sebagai berikut:

```
SELECT
Quarter, SUM(quantity) Total_Penjualan,
SUM(quantity*priceeach) Total_Revenue
FROM
(SELECT *, 1 Quarter FROM
`(project)_customer_report`.pesanan_1
UNION
```

```
SELECT *, 2 Quarter FROM
`(project)_customer_report`.pesanan_2
ORDER BY orderDate ASC) tabel_1
WHERE status='Shipped'
GROUP BY Quarter;
```

Jika *query* tersebut dijalankan, akan menampilkan hasil sebagai berikut:

	Quarter	Total_Penjualan	Total_Revenue
▶	1	8694	799579310
	2	6717	607548320

Gambar 7 Query presentasi Total\_Penjualan dan Total\_Revenue secara keseluruhan

### c. Persentase pertumbuhan

Secara sederhana, pertumbuhan penjualan dan revenue dapat diperoleh dengan mengetahui pertumbuhan penjualan maka dapat dilakukan dengan menghitung persentase penjualan bulan lalu dengan jumlah penjualan sebelumnya dan selisih pendapatan bulan terakhir dengan pendapatan periode sebelumnya (DQ Lab, 2018). Perumusannya seperti berikut ini:

$$\% Y(t) = \frac{\text{jumlah}(t) - \text{jumlah}(t-1)}{\text{jumlah}(t-1)}$$

$$\% X(t) = \frac{\text{jumlah(harga}(t)) - \text{jumlah(harga}(t-1))}{\text{jumlah(harga}(t-1))}$$

Salah satu query yang bisa digunakan adalah sebagai berikut:

```
(SELECT
V, SUM(quantity) Total_Penjualan, SUM(quantity*priceeach)
Total_Revenue
FROM (SELECT *, 'Quarter 1' V FROM
`(project)_customer_report`.pesanan_1
UNION SELECT *, 'Quarter 2' V FROM
`(project)_customer_report`.pesanan_2
ORDER BY orderDate ASC) tabel_1
WHERE status='Shipped'
GROUP BY V)
UNION
(SELECT 'Pertumbuhan (%)',
ROUND
(((Tabel_2.Total_Penjualan-Tabel_1.Total_Penjualan) /
Tabel_1.Total_Penjualan ) * 100,2)
'Pertumbuhan_Total_Penjualan_(%)',
ROUND(((Tabel_2.Total_Revenue-Tabel_1.Total_Revenue) /
Tabel_1.Total_Revenue ) * 100,2)
'Pertumbuhan_Total_Revenue_(%)'
```

```

FROM
(SELECT
0 V, SUM(quantity) Total_Penjualan,
SUM(quantity*priceeach) Total_Revenue
FROM
`(project)_customer_report`.pesanan_1
WHERE status = 'Shipped' ) Tabel_1
INNER JOIN
(SELECT
0 V, SUM(quantity) Total_Penjualan, SUM(quantity*priceeach)
Total_Revenue
FROM
`(project)_customer_report`.pesanan_2
WHERE
status = 'Shipped' ) Tabel_2
ON Tabel_1.V=Tabel_2.V);

```

Jika dijalankan, query tersebut akan memunculkan hasil sebagai berikut:

	V	Total_Penjualan	Total_Revenue
▶	Quarter 1	8694.00	799579310.00
	Quarter 2	6717.00	607548320.00
	Pertumbuhan (%)	-22.74	-24.02

Gambar 8 Output query presentasi Total\_Penjualan dan Total\_Revenue secara keseluruhan

### 3. Analisis Pelanggan

Akan terdapat empat sub-pertanyaan yang akan ditelusuri dalam poin ini. Adapun ke-empat pertanyaan tersebut adalah sebagai berikut:

#### a. Apakah Jumlah pelanggan mengalami penambahan?

Untuk menjawab pertanyaan ini kita dapat meninjaunya dari membandingkan jumlah pelanggan yang melakukan pendaftaran pada periode saat ini dengan total jumlah pelanggan yang melakukan registrasi di akhir periode sebelumnya (DQ Lab, 2018). Secara sederhana diformulasikan sebagai berikut:

$$\% Pgn_{+} = \left( \frac{TPgn_{RSI}}{TPgn_{RSb}} \right) \times 100\%$$

Dengan  $Pgn_{+}$  adalah penambahan pelanggan,  $TPgn_{RSI}$  adalah total pelanggan periode saat ini, dan  $TPgn_{RSb}$  adalah total pelanggan periode sebelumnya.

Salah satu query yang bisa digunakan adalah sebagai berikut ini:

```

SELECT
Quarter, COUNT(DISTINCT (customerID)) Total_Pelanggan
FROM
(SELECT *, QUARTER(createDate) Quarter

```



```

FROM `(project)_customer_report`.pelanggan
WHERE createDate
BETWEEN '2004-01-01' AND '2004-06-30') Tabel_b
GROUP BY Quarter;

```

Hasil yang diperoleh jika menjalankan query tersebut adalah sebagai berikut:

	Quarter	Total_Pelanggan
▶	1	43
	2	35

Gambar 9 Output query penambahan pelanggan

- b. Berapa jumlah pelanggan tersebut yang sudah melakukan transaksi?

Salah satu cara yang bisa dilakukan untuk mengetahuinya adalah dengan menggunakan query berikut:

```

SELECT
Quarter, COUNT(DISTINCT(customerID)) Total_Pelanggan
FROM
(SELECT
*, QUARTER(createDate) Quarter
FROM `(project)_customer_report`.pelanggan
WHERE createDate
BETWEEN '2004-01-01' AND '2004-06-30') Tabel_b
WHERE customerID
IN (SELECT DISTINCT(customerID)
FROM `(project)_customer_report`.pesanan_1
UNION SELECT DISTINCT(customerID)
FROM `(project)_customer_report`.pesanan_2)
GROUP BY Quarter;

```

Jika query tersebut dijalankan, akan menampilkan hasil seperti berikut ini:

	Quarter	Total_Pelanggan
▶	1	25
	2	19

Gambar 10 Output query penambahan pelanggan yang bertransaksi

- c. Kategori produk apa saja yang paling banyak dipesan oleh pelanggan pada quarter 1 dan 2?

Produk yang paling banyak diminati dapat diketahui dengan cara menghitung total pesanan dan jumlah penjualan dari setiap kategori produk. Adapun salah satu query yang dapat digunakan adalah sebagai berikut:

```

SELECT
Tabel_1.categoryID,
Tabel_1.Total_Order Total_Order_Quarter_2,
Tabel_2.Total_Order Total_Order_Quarter_1,
SUM(Tabel_1.Total_Penjualan) Total_Penjualan_Quarter_2,
SUM(Tabel_2.Total_Penjualan) Total_Penjualan_Quarter_1
FROM
( SELECT
categoryID, COUNT(DISTINCT(orderNumber)) Total_Order,
SUM(quantity) Total_Penjualan
FROM
( SELECT
*, LEFT(productCode,3) categoryID
FROM `(project)_customer_report`.pesanan_2
WHERE status = "Shipped" ) Tabel_c
GROUP BY categoryID
ORDER BY Total_Order
DESC ) Tabel_1
INNER JOIN
( SELECT
categoryID, COUNT(DISTINCT(orderNumber)) Total_Order,
SUM(quantity) Total_Penjualan
FROM
( SELECT
*, LEFT(productCode,3) categoryID
FROM `(project)_customer_report`.pesanan_1
WHERE status = "Shipped" ) Tabel_c
GROUP BY categoryID
ORDER BY Total_Order
DESC ) Tabel_2
ON Tabel_1.categoryID = Tabel_2.categoryID
GROUP BY tabel_1.categoryID
ORDER BY Total_Order_Quarter_2
DESC;

```

Hasil yang diperoleh dari query tersebut adalah sebagai berikut ini:

	categoryID	Total_Order_Quarter_2	Total_Order_Quarter_1	Total_Penjualan_Quarter_2	Total_Penjualan_Quarter_1
►	S18	25	24	2264	3026
	S24	21	24	1826	2256
	S32	11	9	616	516
	S12	10	11	491	797
	S10	8	10	492	529
	S50	8	7	292	348
	S70	7	11	675	1001
	S72	2	5	61	221

Gambar 11 Query produk paling laku di quarter 1 dan 2

d. Berapa jumlah pelanggan yang aktif bertransaksi setelah transaksi pertamanya?

Tentunya, mengetahui adakah pelanggan yang masih aktif bertransaksi setelah transaksi pertamanya menjadi informasi penting bagi tim product dan bisnis analis. Karena hal tersebut dapat menjadi tolak ukur apakah produk yang diberikan memuaskan atau tidak dan bahkan bisa menjadi dasar dalam pengembangan produk selanjutnya. Metrik yang biasa digunakan adalah *retention cohort*. Ada beberapa cara untuk menghitung *retention* tersebut diantaranya menggunakan query JOIN dan SELF JOIN. Pun cara lainnya adalah menggunakan query sebagai berikut:

**SELECT**

Tabel\_1.Quarter,

Tabel\_1.Pelanggan\_Q1 Pelanggan\_Q1,

Tabel\_2.Pelanggan\_Q2 Pelanggan\_Q1\_di\_Q2,

**ROUND**((Pelanggan\_Q2 / Pelanggan\_Q1)\*100,2)

'Persentase\_Pelanggan\_Aktif\_ (%)'

**FROM**

( **SELECT 1** Quarter ,

**COUNT**(**DISTINCT**(customerID)) Pelanggan\_Q1

**FROM**

`(project)\_customer\_report`.pesanan\_1 ) Tabel\_1

**INNER JOIN**

( **SELECT 1** Quarter ,

**COUNT**(**DISTINCT**(customerID)) Pelanggan\_Q2

**FROM**

`(project)\_customer\_report`.pesanan\_1

**WHERE** customerID

**IN** ( **SELECT DISTINCT** customerID

**FROM**

`(project)\_customer\_report`.pesanan\_2 ) )

Tabel\_2

**ON** Tabel\_1.Quarter = Tabel\_2.Quarter;

Jika *Query* tersebut dijalankan, akan menampilkan hasil sebagai berikut:

	Quarter	Pelanggan_Q1	Pelanggan_Q1_di_Q2	Persentase_Pelanggan_Aktif_ (%)
►	1	25	6	24.00

Gambar 12 Output query pelanggan yang aktif bertransaksi setelah transaksi pertamanya

## Hasil

Melihat hasil yang telah diperoleh, dapat ditarik beberapa benang merah diantaranya adalah terjadinya penurunan performa yang signifikan di quarter ke-2 dengan nilai penurunan penjualan dan revenue masing-masing adalah 22% dan 24%. Hal tersebut diduga selaras dengan terjadinya penurunan jumlah pembeli di quarter-2 yang mana pelanggan juga mengalami penurunan sebanyak 8 orang. Untuk itu kepada tim produksi, diharapkan perlu mempelajari kembali terkait kebiasaan pelanggan dan melakukan *improvement* terhadap produk yang ingin disediakan.

Melihat dari sisi produk, kategori produk dengan kode S18 dan S24 memiliki nilai penjualan tertinggi, dengan kontribusi pesanan mencapai 50% dan total penjualan mencapai 60%. Hal ini menjadi informasi bagus untuk kinerja tim, dan sebaiknya untuk periode kedepannya, penjualan difokuskan pada dua produk ini.

Berdasarkan perilaku pelanggan yang aktif bertransaksi kembali setelah transaksi pertamanya, nilai *retention rate* diperoleh sebesar 24%, yang mana tergolong rendah untuk suatu bisnis. Perlu diketahui bahwa *Retention rate* sendiri merupakan salah satu cara sederhana mengetahui kesetiaan pelanggan menggunakan jasa suatu perusahaan, bisa berarti mencerminkan kepuasan pelanggan itu sendiri (Gie, 2021). Nilai yang rendah menjadi indikator bahwa terdapat ketidakpuasan pelanggan dalam menggunakan produk dari suatu perusahaan tersebut. Untuk mencegah hal ini terjadi lagi di quarter selanjutnya, disarankan perusahaan mengatur strategi dalam melakukan banyak improvisasi baik dari segi produk ataupun marketing bisnis.

## Referensi

- Andry, T. (2018, Maret 28). *Cara Mengambil Keputusan*. Dipetik Mei 25, 2022, dari <https://entrepreneurcamp.id>: <https://entrepreneurcamp.id/cara-mengambil-keputusan/>
- DQ Lab. (2018). *Apakah jumlah customers xyz.com semakin bertambah?* Dipetik Mei 24, 2022, dari <https://academy.dqlab.id>: <https://academy.dqlab.id/main/livecode/246/421/2106?pr=0>
- DQ Lab. (2018). *Project Data Analysis for B2B Retail: Customer Analytics Report*. Dipetik Mei 24, 2022, dari <https://academy.dqlab.id>: <https://academy.dqlab.id/main/package/project/246>
- DQ Lab. (2018). *Tabel yang Digunakan*. Dipetik Mei 24, 2022, dari <https://academy.dqlab.id>: <https://academy.dqlab.id/main/projectcode/246/418/2101?pr=>
- Gie. (2021, Juni 4). *Apa itu tingkat retensi pelanggan?* Dipetik Mei 25, 2022, dari <https://accurate.id>: <https://accurate.id/marketing-manajemen/apa-itu-tingkat-retensi-pelanggan/>
- Ibnu. (2020, Desember 28). *revenue-adalah*. Dipetik Mei 24, 2022, dari <https://accurate.id>: <https://accurate.id/akuntansi/revenue-adalah/>