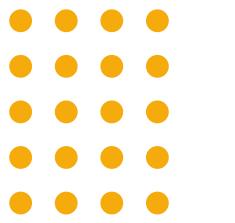


**PORTFOLIO DATA ANALYST**



# **USER RETENTION ANALYSIS**

**Kayla Alysa Adra**

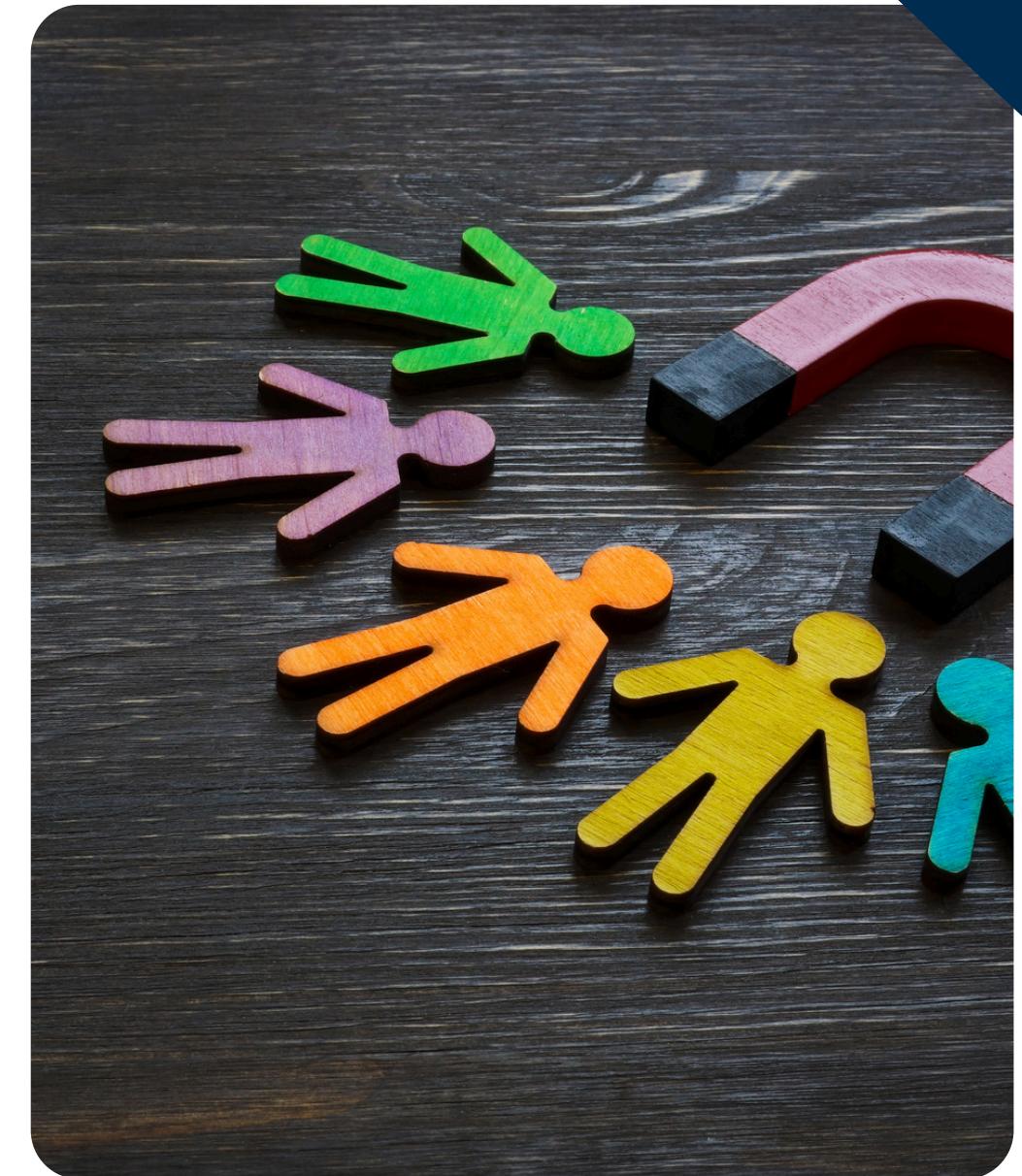


# USER RETENTION



User Retention adalah metrik yang mengukur persentase pengguna yang tetap aktif dalam jangka waktu tertentu setelah pertama kali menggunakan layanan atau produk.

$$\text{Retention rate} = \frac{\text{Banyaknya pengguna yang kembali bertransaksi}}{\text{Banyaknya pengguna yang sebelumnya telah bertransaksi}}$$



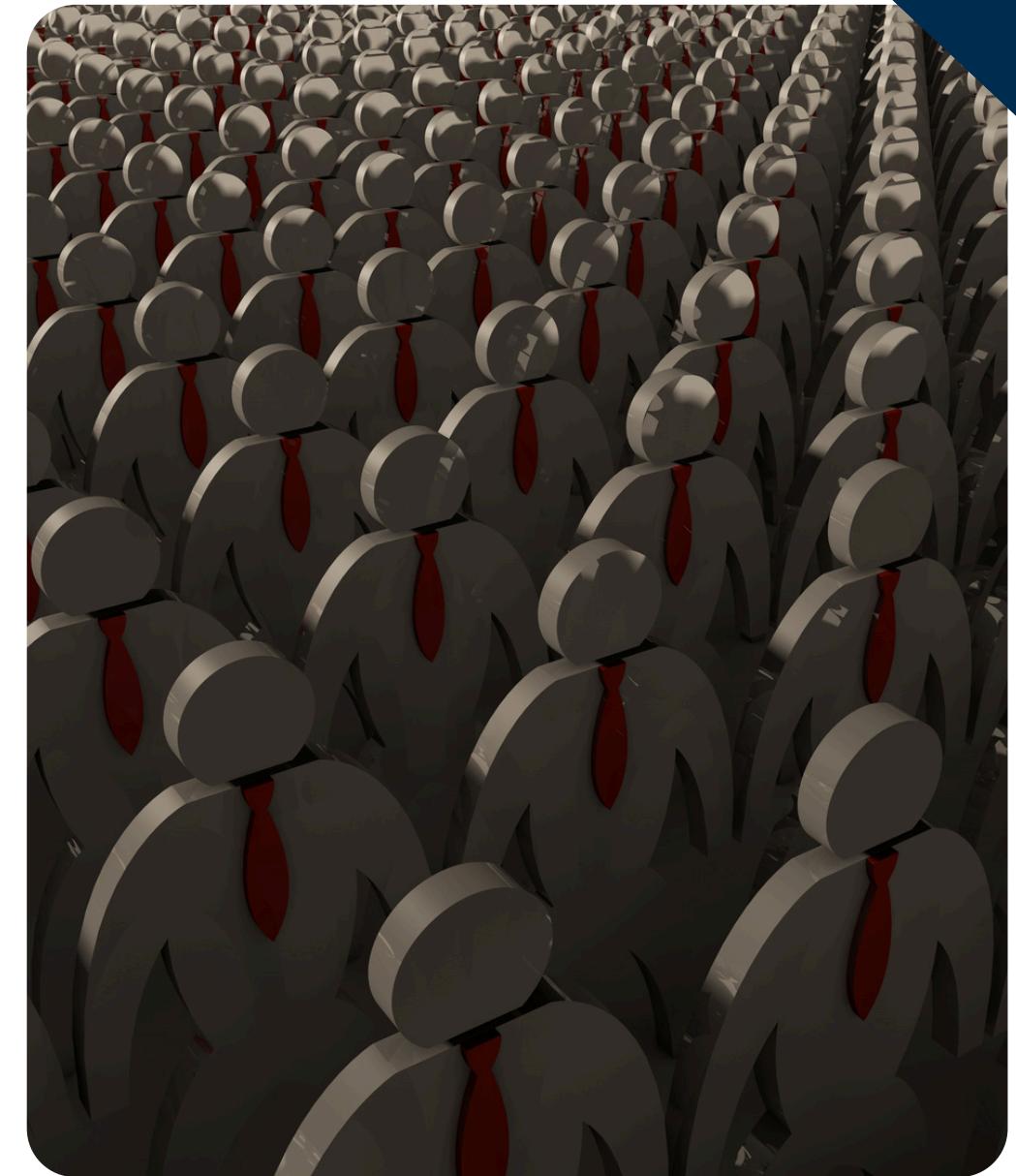
# COHORT ANALYST



Cohort analysis adalah teknik yang digunakan untuk menganalisis dan memahami perilaku sekelompok individu dari waktu ke waktu.

Manfaat:

- Memahami tren retensi pengguna dalam jangka panjang.
- Mengetahui periode waktu di mana pengguna mulai churn (berhenti menggunakan produk).
- Mengevaluasi efektivitas strategi pemasaran atau fitur baru.



# DATA ONLINE RETAIL



Columns	Dtypes
Order_id	object
product_code	object
product_name	object
quantity	int64
order_date	object
price	float64
customer_id	float64

	order_id	product_code	product_name	quantity	order_date	price	customer_id
0	493410	TEST001	This is a test product.	5	2010-01-04 09:24:00	4.50	12346.0
1	C493411	21539	RETRO SPOTS BUTTER DISH	-1	2010-01-04 09:43:00	4.25	14590.0
2	493412	TEST001	This is a test product.	5	2010-01-04 09:53:00	4.50	12346.0
3	493413	21724	PANDA AND BUNNIES STICKER SHEET	1	2010-01-04 09:54:00	0.85	NaN
4	493413	84578	ELEPHANT TOY WITH BLUE T-SHIRT	1	2010-01-04 09:54:00	3.75	NaN
...	...	...	...	...	...	...	...
461768	539991	21618	4 WILDFLOWER BOTANICAL CANDLES	1	2010-12-23 16:49:00	1.25	NaN
461769	539991	72741	GRAND CHOCOLATECANDLE	4	2010-12-23 16:49:00	1.45	NaN
461770	539992	21470	FLOWER VINE RAFFIA FOOD COVER	1	2010-12-23 17:41:00	3.75	NaN
461771	539992	22258	FELT FARM ANIMAL RABBIT	1	2010-12-23 17:41:00	1.25	NaN
461772	539992	21155	RED RETROSPOT PEG BAG	1	2010-12-23 17:41:00	2.10	NaN

461773 rows × 7 columns

# STEP BY STEP



01

Import Libraries

03

Data Cleaning

02

Import Dataset

04

Buat User Retention Cohort

# IMPORT LIBRARIES

```
import pandas as pd
import numpy as np
import datetime as dt
from scipy import stats
from operator import attrgetter
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import seaborn as sns
```

1. **pandas** – Untuk memproses dan menganalisis data dalam bentuk tabel (DataFrame).
2. **numpy** – Untuk operasi numerik dan manipulasi array.
3. **datetime** – Untuk menangani data waktu dan tanggal.
4. **scipy.stats** – Untuk perhitungan statistik tambahan seperti uji distribusi data & menghapus outlier.
5. **operator.attrgetter** – Untuk mendapatkan atribut dari objek dengan lebih efisien.
6. **matplotlib.pyplot** – Untuk membuat visualisasi data dalam bentuk grafik.
7. **matplotlib.colors** – Untuk mengatur warna dalam visualisasi.
8. **seaborn** – Untuk visualisasi data yang lebih menarik dan informatif.

# IMPORT DATAFRAME

```
df = pd.read_csv('Salinan Online Retail Data.csv', header=0)
```

	order_id	product_code	product_name	quantity	order_date	price	customer_id
0	493410	TEST001	This is a test product.	5	2010-01-04 09:24:00	4.50	12346.0
1	C493411	21539	RETRO SPOTS BUTTER DISH	-1	2010-01-04 09:43:00	4.25	14590.0
2	493412	TEST001	This is a test product.	5	2010-01-04 09:53:00	4.50	12346.0
3	493413	21724	PANDA AND BUNNIES STICKER SHEET	1	2010-01-04 09:54:00	0.85	NaN
4	493413	84578	ELEPHANT TOY WITH BLUE T-SHIRT	1	2010-01-04 09:54:00	3.75	NaN
...	...	...	...	...	...	...	...
461768	539991	21618	4 WILDFLOWER BOTANICAL CANDLES	1	2010-12-23 16:49:00	1.25	NaN
461769	539991	72741	GRAND CHOCOLATECANDLE	4	2010-12-23 16:49:00	1.45	NaN
461770	539992	21470	FLOWER VINE RAFFIA FOOD COVER	1	2010-12-23 17:41:00	3.75	NaN
461771	539992	22258	FELT FARM ANIMAL RABBIT	1	2010-12-23 17:41:00	1.25	NaN
461772	539992	21155	RED RETROSPOT PEG BAG	1	2010-12-23 17:41:00	2.10	NaN

461773 rows × 7 columns

# DATA CLEANING

01

Membuat salinan data

```
df_clean = df.copy()
```

02

Mengonversi kolom order\_date menjadi tipe datetime

```
df_clean['order_date'] = df_clean['order_date'].astype('datetime64[ns]')
```

# DATA CLEANING

03

## Membuat kolom year\_month

Kolom ini berisi periode dalam format tahun dan bulan, yang berguna dalam analisis tren bulanan.

```
df_clean['year_month'] = df_clean['order_date'].dt.to_period('M')
```

04

## Menghapus baris tanpa customer\_id

Baris yang tidak memiliki customer\_id dihapus karena tidak dapat digunakan dalam analisis pelanggan.

```
df_clean = df_clean[~df_clean['customer_id'].isna()]
```

# DATA CLEANING

05

## Menghapus baris tanpa product\_name

Baris yang tidak memiliki informasi nama produk juga dihapus agar analisis lebih akurat.

```
df_clean = df_clean[~df_clean['product_name'].isna()]
```

06

## Mengubah semua product\_name menjadi huruf kecil

Ini bertujuan untuk menyamakan format nama produk agar tidak ada perbedaan akibat perbedaan huruf kapital.

```
df_clean['product_name'] = df_clean['product_name'].str.lower()
```

# DATA CLEANING

07

Menghapus semua baris dengan `product_code` atau `product_name` yang mengandung kata 'test'

```
df_clean = df_clean[~df_clean['product_code'].str.lower().str.contains('test')) |  
                    (~df_clean['product_name'].str.contains('test '))]
```

08

Menentukan `order_status` berdasarkan `order_id`

Jika `order_id` diawali dengan huruf 'C', status pesanan dianggap "cancelled". Jika tidak, dianggap "delivered".

```
df_clean['order_status'] = np.where(df_clean['order_id'].str[:1]=='C', 'cancelled', 'delivered')
```

# DATA CLEANING

09

## Mengubah nilai quantity yang negatif menjadi positif

Nilai negatif hanya menandakan pesanan yang dibatalkan, sehingga dikonversi menjadi positif.

```
df_clean['quantity'] = df_clean['quantity'].abs()
```

10

## Menghapus baris dengan price bernilai negatif

Harga yang negatif tidak valid dan harus dihapus.

```
df_clean = df_clean[df_clean['price'] > 0]
```

# DATA CLEANING

11

## Membuat kolom amount

Amount dihitung sebagai hasil perkalian antara quantity dan price.

```
df_clean['amount'] = df_clean['quantity'] * df_clean['price']
```

12

## Menyesuaikan product\_name berdasarkan product\_code

Jika satu kode produk memiliki beberapa nama produk, digunakan nama produk yang paling sering muncul.

# DATA CLEANING

13

Mengonversi `customer_id` menjadi string

```
df_clean['customer_id'] = df_clean['customer_id'].astype(str)
```

14

Menghapus Outlier

Outlier dihapus berdasarkan nilai z-score untuk kolom quantity dan amount dengan batas 3 standar deviasi.

```
df_clean = df_clean[(np.abs(stats.zscore(df_clean[['quantity', 'amount']]))<3).all(axis=1)]
df_clean = df_clean.reset_index(drop=True)
df_clean
```



# USER RETENTION COHORT



Agregasi data transaksi ke bentuk summary total transaksi per pengguna setiap bulan

```
df_user_monthly = df_clean.groupby(['customer_id', 'year_month'],
                                   as_index=False).agg(order_cnt=('order_id','nunique'))
```

	customer_id	year_month	order_cnt
0	12346.0	2010-01	1
1	12346.0	2010-03	1
2	12346.0	2010-06	2
3	12346.0	2010-10	1
4	12608.0	2010-10	1
...	...	...	...
12039	18286.0	2010-06	1
12040	18286.0	2010-08	1
12041	18287.0	2010-05	1
12042	18287.0	2010-09	2
12043	18287.0	2010-11	1

Langkah ini mengelompokkan data berdasarkan customer\_id dan year\_month serta menghitung jumlah order unik setiap kombinasi

# USER RETENTION COHORT

Menambahkan kolom cohort yang menunjukkan bulan pertama kali setiap pengguna bertransaksi

```
df_user_monthly['cohort'] = df_user_monthly.groupby('customer_id')['year_month'].transform('min')
```

	customer_id	year_month	order_cnt	cohort
0	12346.0	2010-01	1	2010-01
1	12346.0	2010-03	1	2010-01
2	12346.0	2010-06	2	2010-01
3	12346.0	2010-10	1	2010-01
4	12608.0	2010-10	1	2010-10
...	...	...	...	...
12039	18286.0	2010-06	1	2010-06
12040	18286.0	2010-08	1	2010-06
12041	18287.0	2010-05	1	2010-05
12042	18287.0	2010-09	2	2010-05
12043	18287.0	2010-11	1	2010-05

# USER RETENTION COHORT

Menghitung jarak (dalam bulan) antara transaksi saat ini dengan bulan pertama kali transaksi (cohort)

```
df_user_monthly['period_num'] = (df_user_monthly['year_month'] -  
                                 df_user_monthly['cohort']).apply(attrgetter('n')) + 1  
df_user_monthly
```

	customer_id	year_month	order_cnt	cohort	period_num
0	12346.0	2010-01	1	2010-01	1
1	12346.0	2010-03	1	2010-01	3
2	12346.0	2010-06	2	2010-01	6
3	12346.0	2010-10	1	2010-01	10
4	12608.0	2010-10	1	2010-10	1
...	...	...	...	...	...
12039	18286.0	2010-06	1	2010-06	1
12040	18286.0	2010-08	1	2010-06	3
12041	18287.0	2010-05	1	2010-05	1
12042	18287.0	2010-09	2	2010-05	5
12043	18287.0	2010-11	1	2010-05	7



# USER RETENTION COHORT



Membuat tabel pivot yang menampilkan jumlah pengguna unik  
Tabel pivot dibuat dengan index berupa cohort, kolom berupa periode (jarak bulan), dan nilai yang merupakan jumlah unik customer\_id.

```
df_cohort_pivot = pd.pivot_table(df_user_monthly, index='cohort',
                                  columns='period_num', values='customer_id', aggfunc=pd.Series.nunique)
df_cohort_pivot
```

cohort	period_num	1	2	3	4	5	6	7	8	9	10	11	12
	2010-01	713.0	280.0	334.0	313.0	305.0	304.0	293.0	268.0	285.0	319.0	335.0	249.0
2010-02	461.0	154.0	128.0	161.0	152.0	121.0	119.0	159.0	153.0	166.0	100.0	NaN	
2010-03	528.0	146.0	158.0	145.0	140.0	123.0	149.0	186.0	193.0	96.0	NaN	NaN	
2010-04	326.0	82.0	75.0	63.0	69.0	79.0	98.0	101.0	50.0	NaN	NaN	NaN	
2010-05	274.0	55.0	50.0	52.0	52.0	72.0	67.0	43.0	NaN	NaN	NaN	NaN	
2010-06	266.0	53.0	56.0	60.0	65.0	85.0	39.0	NaN	NaN	NaN	NaN	NaN	
2010-07	179.0	38.0	37.0	52.0	53.0	30.0	NaN	NaN	NaN	NaN	NaN	NaN	
2010-08	160.0	35.0	50.0	48.0	27.0	NaN							
2010-09	227.0	64.0	60.0	31.0	NaN								
2010-10	362.0	103.0	61.0	NaN									
2010-11	327.0	66.0	NaN										
2010-12	66.0	NaN											

Tabel ini akan menampilkan berapa banyak pengguna dari tiap cohort yang melakukan transaksi pada periode tertentu.

# USER RETENTION COHORT

# Menghitung Retention Rate

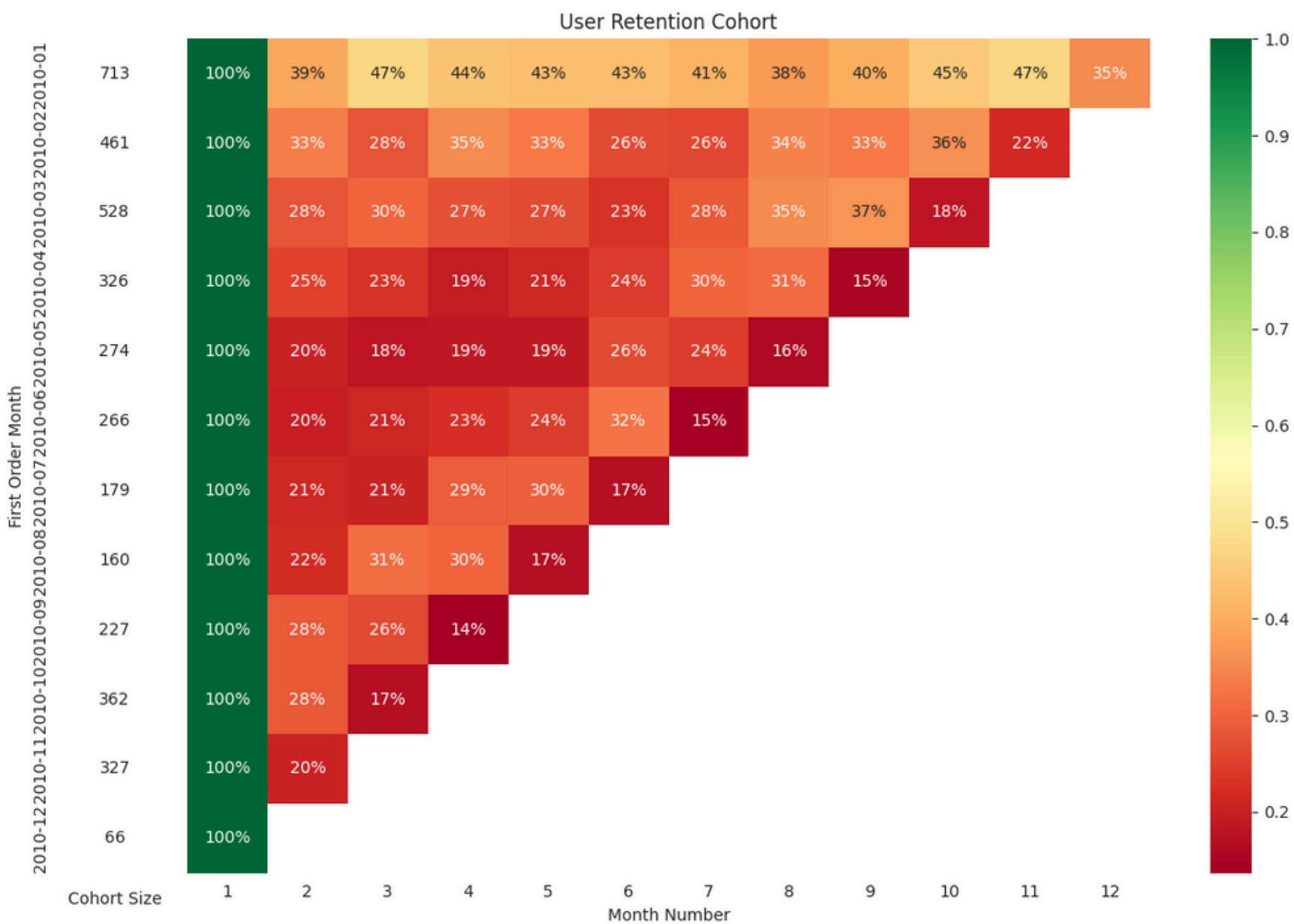
- Langkah pertama adalah menentukan ukuran masing-masing cohort dengan mengambil nilai pada periode pertama
  - Selanjutnya, setiap nilai pada tabel pivot dibagi dengan ukuran cohort-nya untuk mendapatkan retention rate

```
cohort_size = df_cohort_pivot.iloc[:,  
cohort_size
```

```
cohort
2010-01      713.0
2010-02      461.0
2010-03      528.0
2010-04      326.0
2010-05      274.0
2010-06      266.0
2010-07      179.0
2010-08      160.0
2010-09      227.0
2010-10      362.0
2010-11      327.0
2010-12       66.0
Freq: M, Name: 1, dtype: float64
```

```
df_retention_cohort = df_cohort_pivot.divide(cohort_size, axis=0)
df_retention_cohort
```

# USER RETENTION COHORT



Beberapa insight yang dapat diperoleh dari analisis ini antara lain:

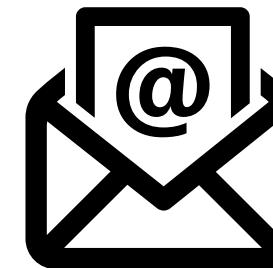
- Pengguna paling banyak pertama kali bertransaksi pada Januari 2010 (713 pengguna).
- Cohort pengguna tersebut juga yang paling banyak bertransaksi kembali di bulan kedua dengan retention rate sebesar 39% dibanding cohort lain.
- Cohort tersebut menunjukkan loyalitas yang lebih tinggi di bulan-bulan berikutnya dengan retention rate berkisar di atas 40%.
- Namun, secara keseluruhan, sebagian besar pengguna tidak kembali bertransaksi karena banyak cohort yang memiliki retention rate di bawah 50%.
- Retention rate pada Desember 2010 tampak paling rendah dibandingkan dengan bulan-bulan sebelumnya, yang mengindikasikan penurunan aktivitas pengguna di akhir tahun.

# TERIMA KASIH



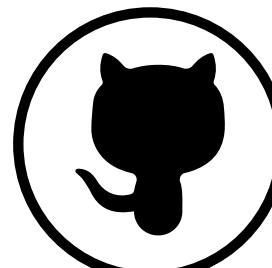
**LINKEDIN**

[www.linkedin.com/in/kaylaalysa](https://www.linkedin.com/in/kaylaalysa)



**EMAIL**

[kaylaaadra12@gmail.com](mailto:kaylaaadra12@gmail.com)



**GITHUB**

<https://github.com/kaylaalysa>