

# Building a Customer-Centric Retail Data Mart for Behavioral Analysis

Muhammad Arif Nur Sidik

---

<https://github.com/arifnrsk/Building-a-Customer-Centric-Retail-Data-Mart-for-Behavioral-Analysis>



# Muhammad Arif Nur Sidik

## Education

*SI Computer Science | Binus University*

## Working

*Students*

## Overview Project

- **Building a Customer-Centric Retail Data Mart for Behavioral Analysis**

Membangun data insights platform untuk mengubah data transaksi menjadi insight bisnis mendalam, dengan melakukan analisis pola belanja (Market Basket Analysis) dan segmentasi perilaku pelanggan (RFM), guna mengoptimalkan strategi penjualan dan marketing yang lebih tertarget.

On the left side of the slide, there are three overlapping geometric shapes: a large black parallelogram at the top, a medium-sized light orange parallelogram in the middle, and a smaller dark orange parallelogram at the bottom. All shapes are slanted to the right.


# Project Background

## Project Background

Banyak toko ritel semi modern menghasilkan data transaksi harian yang signifikan. Namun, data ini seringkali belum dimanfaatkan secara maksimal untuk memahami pendorong utama bisnis, yaitu perilaku dan pola belanja pelanggan.

Akibatnya, keputusan strategis seperti penataan letak produk, promosi bundling, dan program loyalitas pelanggan seringkali didasarkan pada intuisi, bukan pada data yang terukur.

Tujuan utama project ini adalah membangun sebuah data insights platform yang mampu mengolah raw data transaction menjadi dua insight yaitu analisis pola belanja (Market Basket Analysis) dan segmentasi perilaku pelanggan (RFM), untuk mendukung pengambilan keputusan yang sepenuhnya data-driven.

A large black parallelogram is positioned on the left side of the slide. Overlapping its bottom edge are two smaller parallelograms: a light orange one in front and a darker orange one behind it.

# Problem Statement

# Problem Statement

## 1. Kurangnya pemahaman tentang perilaku belanja individu

Mengakibatkan hilangnya kesempatan untuk membangun loyalitas dan melakukan marketing yang tertarget ke segmen pelanggan paling berpotensi.

## 2. Penataan produk yang tidak optimal

Penataan letak produk di toko dan strategi promosi bundling seringkali didasarkan pada kebiasaan, bukan pada data pola belanja pelanggan. Akibatnya, potensi cross-selling untuk produk yang sering dibeli bersamaan menjadi tidak maksimal.

# Problem Statement

## 3. Keterbatasan aplikasi PoS untuk analisis

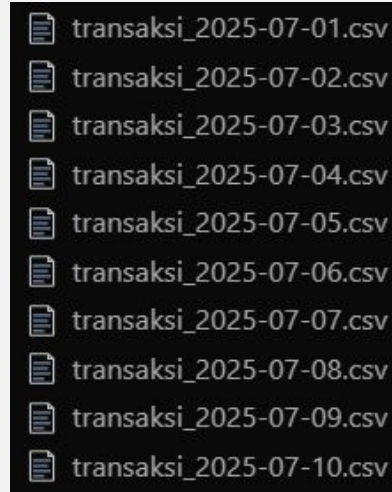
Sistem PoS standar dioptimalkan untuk kecepatan transaksi, bukan untuk melakukan analisis perilaku yang kompleks seperti Market Basket Analysis dan segmentasi RFM.

A large black parallelogram is positioned on the left side of the slide. Below it, two overlapping parallelograms in shades of orange and yellow are also positioned on the left, creating a layered, geometric effect.

# Data Platform Understanding



## Data Source



Platform ini menggunakan data yang di-generate secara lokal menggunakan script Python. Tujuannya adalah untuk mensimulasikan data transaksi harian yang dinamis, untuk membuat kasus penggunaan pipeline lebih realistis.

Setiap file CSV merepresentasikan data transaksi untuk satu hari spesifik.

# Data Source

	A	B	C	D	E	F	G
1	StockCode	Description	Price	Quantity	Invoice	InvoiceDate	Customer ID
2	SNK-002	Chitato Sapi Panggang	11000	3	INV-20240101-1	2024-01-01	12902.0
3	HHLD-001	Sunlight Pencuci Piring	14000	1	INV-20240101-2	2024-01-01	14088.0
4	HHLD-001	Sunlight Pencuci Piring	14000	5	INV-20240101-3	2024-01-01	13563.0
5	HHLD-002	Spon Cuci Piring	2000	4	INV-20240101-4	2024-01-01	15974.0
6	SNK-002	Chitato Sapi Panggang	11000	1	INV-20240101-4	2024-01-01	15974.0
7	IND-002	Telur Ayam (1 butir)	2500	4	INV-20240101-4	2024-01-01	
8	BEV-001	Teh Botol Sosro Kotak	3500	5	INV-20240101-4	2024-01-01	15974.0
9	IND-002	Telur Ayam (1 butir)	2500	1	INV-20240101-5	2024-01-01	17214.0
10	IND-002	Telur Ayam (1 butir)	2500	4	INV-20240101-5	2024-01-01	17214.0
11	BEV-001	Teh Botol Sosro Kotak	3500	5	INV-20240101-6	2024-01-01	14791.0
12	BEV-001	Teh Botol Sosro Kotak	3500	4	INV-20240101-6	2024-01-01	14791.0
13	IND-001	Indomie Goreng Original	3000	1	INV-20240101-6	2024-01-01	14791.0
14	BEV-001	Teh Botol Sosro Kotak	3500	2	INV-20240101-6	2024-01-01	14791.0
15	IND-002	Telur Ayam (1 butir)	2500	3	INV-20240101-7	2024-01-01	13676.0
16	HHLD-001	Sunlight Pencuci Piring	14000	1	INV-20240101-7	2024-01-01	13676.0
17	IND-003	Saus Sambal Botol	8000	2	INV-20240101-7	2024-01-01	13676.0
18	IND-003	Saus Sambal Botol	0	3	INV-20240101-8	2024-01-01	17973.0
19	IND-002	Telur Ayam (1 butir)	2500	2	INV-20240101-8	2024-01-01	17973.0
20	RICE-001	Beras 5kg	65000	5	INV-20240101-9	2024-01-01	14766.0

# Data Processing and Storage

Raw data akan diolah dan ditransformasi menggunakan Apache Spark (PySpark) untuk melakukan dua analisis utama:

- **Market Basket Analysis (MBA)** untuk menemukan menemukan produk yang sering dibeli bersamaan.
- **Segmentasi Pelanggan (RFM)** untuk mengelompokkan pelanggan berdasarkan perilaku.

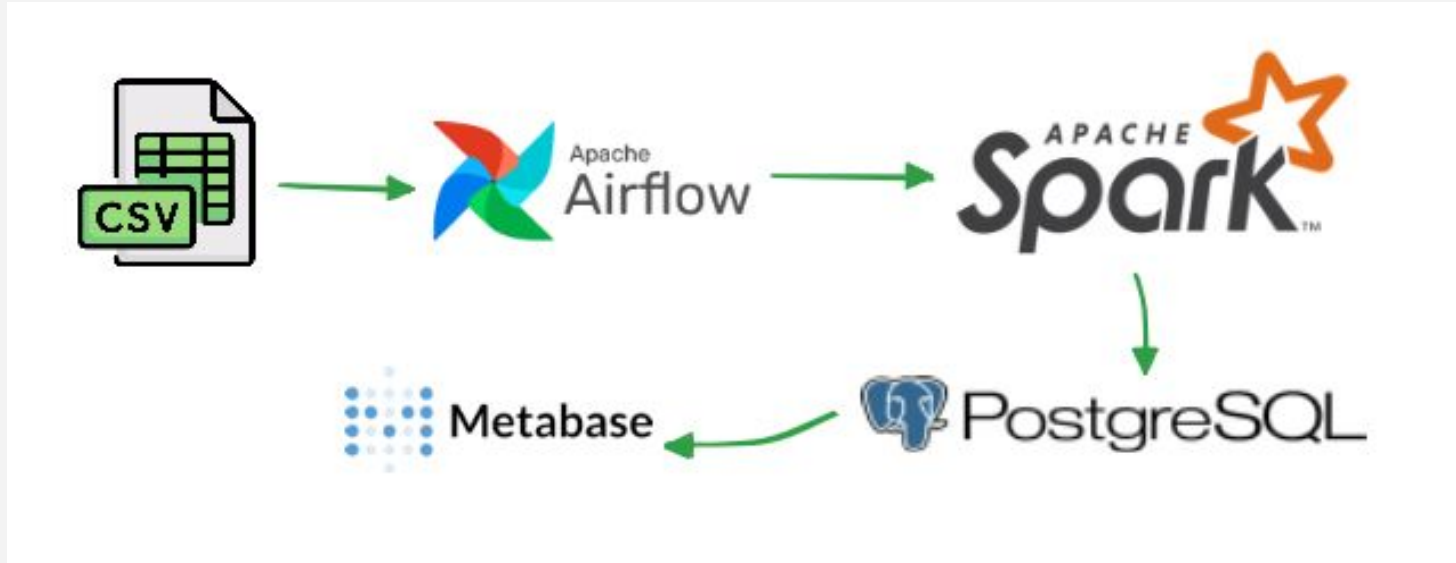
# Visualization & Monitoring

Data dari PostgreSQL akan dihubungkan ke dashboard interaktif yang dibangun menggunakan Metabase.

Dashboard ini akan menyajikan informasi seperti:

- Rekomendasi penataan produk berdasarkan asosiasi barang (hasil MBA).
- Identifikasi segmen pelanggan (seperti 'Loyal Customers' atau 'New Customers') untuk keperluan marketing yang tertarget (hasil RFM).

## Data Flow Diagram



Generated data (CSV) → Python/Airflow (Scheduler) → PySpark (Transformation) → PostgreSQL (2 Tables) → Metabase (Visualization)

A large black parallelogram is positioned on the left side of the slide. Overlapping its bottom edge are two orange parallelograms, one in a lighter shade and one in a darker shade, creating a layered effect.

# Data Understanding

## Data Sources and Characteristics

Data disimulasikan menggunakan script Python (`generate_data.py`) untuk menciptakan data transaksi harian yang dinamis dan realistis, menjawab kebutuhan project akan sumber data yang tidak statis.

Total 1.000.000 baris data transaksi dihasilkan, yang dipecah ke dalam file-file CSV harian untuk mensimulasikan data yang masuk secara berkala.

Kolom dan tipe data dirancang menyerupai data transaksi ritel pada umumnya, mencakup:

- Invoice
- StockCode
- Quantity
- Price
- Customer ID

# Data Quality and Anomaly Simulation

Untuk memastikan pipeline data yang dibangun reliabel, data sengaja di-generate dengan beberapa anomali yang umum ditemukan secara real:

- **Item Non-Produk**  
Sejumlah kecil transaksi (~1%) dibuat untuk item non-produk (misal: 'BIAYA-ADMIN', 'ONGKIR') untuk menguji filtering logic pada tahap transformation.
- **Data Tidak Valid**  
Anomali seperti Price bernilai 0 dan Customer ID yang kosong (null) juga dimasukkan secara acak untuk memastikan proses data cleaning dapat menanganinya dengan benar.



A large black parallelogram is positioned on the left side of the slide. Overlapping its bottom edge are two orange parallelograms, one in a lighter shade and one in a darker shade, both pointing towards the right.

# Transformation & Consideration

# Data Loading & Cleaning with PySpark

Mengambil semua raw transaction file harian dan melakukan data cleaning. Setelahnya dataframe akan digunakan untuk semua analisis selanjutnya.

- **Data ingestion**  
Job Spark dirancang untuk membaca semua file CSV harian dari direktori data sekaligus menggunakan wildcard path (\*.csv). Ini memastikan pipeline memproses semua data yang available.
- **Data cleaning**  
Untuk memastikan data integrity, diimplementasi beberapa filter untuk menghapus data yang "kotor". Ini termasuk memfilter item non-produk (misal: 'BIAYA-ADMIN') dan transaksi dengan harga tidak valid (misal: harga 0).

# Market Basket Analysis

Untuk menemukan produk yang sering dibeli bersamaan, data dikelompokkan berdasarkan Invoice untuk membuat "keranjang belanja".

Kemudian, algoritma FP-Growth dari Spark digunakan untuk menganalisis keranjang ini dan menghasilkan aturan asosiasi (misal: "Jika pelanggan membeli produk A, maka ada kemungkinan X% mereka juga membeli produk B").

# Market Basket Analysis Code

```
print("Performing Market Basket Analysis...")
baskets_df = cleaned_df.groupBy("Invoice").agg(collect_set("StockCode").alias("items"))
baskets_df = baskets_df.filter(size(col("items")) >= 2)

fpGrowth = FPGrowth(itemsCol="items", minSupport=0.005, minConfidence=0.05)
model = fpGrowth.fit(baskets_df)
association_rules = model.associationRules

# Join with product lookup to get antecedent and consequent names
mba_with_antecedent_names = association_rules.withColumn("antecedent_item", explode(col("antecedent"))) \
    .join(product_lookup, col("antecedent_item") == product_lookup.StockCode) \
    .groupBy(association_rules.columns) \
    .agg(collect_set("Description").alias("antecedent_names"))

mba_final_results = mba_with_antecedent_names.withColumn("consequent_item", explode(col("consequent"))) \
    .join(product_lookup, col("consequent_item") == product_lookup.StockCode) \
    .groupBy(mba_with_antecedent_names.columns) \
    .agg(collect_set("Description").alias("consequent_names"))

print("Market Basket Analysis complete. Sample association rules with names:")
mba_final_results.show(5, truncate=False)
```

# Market Basket Analysis Metabase Sample Result

Behavioral Analysis / Market Basket Analysis

Antecedent Names	Consequent Names	Confidence	Lift	Support	+
["Teh Botol Sosro Kotak"; "Sunlight Pencuci Piring"; "Aqua Botol 600ml"]	["Spon Cuci Piring"]	0.62	1.7	0.013	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"; "Sunlight Pencuci Piring"]	["Aqua Botol 600ml"]	0.53	1.55	0.013	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Aqua Botol 600ml"]	0.54	1.56	0.033	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Sunlight Pencuci Piring"]	0.41	1.6	0.025	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Indomie Goreng Original"]	0.091	0.35	0.0056	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Telur Ayam (1 butir)"]	0.16	0.41	0.0099	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Saus Sambal Botol"]	0.096	0.43	0.0059	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Beras 5kg"]	0.095	0.42	0.0059	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Qtela Keripik Singkong"]	0.095	0.42	0.0059	
["Spon Cuci Piring"; "Teh Botol Sosro Kotak"]	["Chitato Sapi Panggang"]	0.096	0.42	0.0059	
["Teh Botol Sosro Kotak"; "Indomie Goreng Original"]	["Aqua Botol 600ml"]	0.54	1.56	0.021	
["Teh Botol Sosro Kotak"; "Indomie Goreng Original"]	["Spon Cuci Piring"]	0.14	0.39	0.0056	
["Teh Botol Sosro Kotak"; "Indomie Goreng Original"]	["Telur Ayam (1 butir)"]	0.72	1.84	0.028	
["Teh Botol Sosro Kotak"; "Aqua Botol 600ml"; "Indomie Goreng Original"]	["Telur Ayam (1 butir)"]	0.71	1.83	0.015	
["Teh Botol Sosro Kotak"; "Indomie Goreng Original"; "Telur Ayam (1 butir)"]	["Aqua Botol 600ml"]	0.53	1.55	0.015	

## **RFM (Recency, Frequency, and Monetary Value)**

**Secara paralel, dilakukan segmentasi pelanggan menggunakan model RFM. Untuk setiap Customer ID, dihitung Recency (kapan terakhir kali belanja), Frequency (seberapa sering belanja), dan Monetary (total nilai belanja).**

**Hasilnya adalah pengelompokan pelanggan yang bisa digunakan untuk strategi marketing yang tertarget.**

# RFM Code

```
print("Performing RFM Analysis...")
df_with_total_price = cleaned_df.withColumn("TotalPrice", col("Quantity") * col("Price"))
snapshot_date = df_with_total_price.agg(_max(col("InvoiceDate"))).first()[0]

rfm_calculated_df = df_with_total_price.groupBy("Customer ID").agg(
    datediff(lit(snapshot_date), _max(col("InvoiceDate"))).alias("Recency"),
    countDistinct("Invoice").alias("Frequency"),
    _sum("TotalPrice").alias("Monetary")
).filter(col("Customer ID").isNotNull())

r_window = Window.orderBy(col("Recency").desc())
f_window = Window.orderBy(col("Frequency"))
m_window = Window.orderBy(col("Monetary"))

rfm_with_scores = rfm_calculated_df.withColumn("r_score", ntile(5).over(r_window)) \
    .withColumn("f_score", ntile(5).over(f_window)) \
    .withColumn("m_score", ntile(5).over(m_window))

rfm_final_df = rfm_with_scores.withColumn("customer_segment",
    when((col("r_score") >= 4) & (col("f_score") >= 4), "Champions")
    .when((col("r_score") >= 4) & (col("f_score") >= 2), "Potential Loyalists")
    .when((col("r_score") >= 3) & (col("f_score") >= 3), "Loyal Customers")
    .when((col("r_score") <= 2) & (col("f_score") >= 3), "At Risk")
    .when((col("r_score") >= 3) & (col("f_score") <= 2), "New Customers")
    .otherwise("Needs Attention")
)

print(f"RFM analysis completed for {rfm_final_df.count()} customers. Sample segments:")
rfm_final_df.show(5)
```

# RFM Metabase Sample Result

Behavioral Analysis / Customer Segmentation					
Customer ID	Recency	Frequency	Monetary	Customer Segment	+
14,167	19	41	3,111,500	Needs Attention	
14,422	22	43	2,451,500	Needs Attention	
17,343	126	44	3,756,000	Needs Attention	
13,352	55	44	2,673,500	Needs Attention	
16,574	3	44	3,275,000	New Customers	
15,155	47	45	3,071,000	Needs Attention	
15,523	11	45	2,589,000	New Customers	
14,469	1	45	2,850,000	New Customers	
13,940	21	46	2,440,000	Needs Attention	
17,021	13	46	1,991,500	Needs Attention	
14,411	4	46	2,929,500	New Customers	
17,462	32	47	3,554,000	Needs Attention	
12,621	24	47	3,046,000	Needs Attention	
14,204	21	47	3,312,000	Needs Attention	
14,876	17	47	2,392,500	Needs Attention	



## Loading to PostgreSQL Data Mart

Load hasil analisis ke dalam tempat data storage, yaitu PostgreSQL dimana berfungsi sebagai Data Mart.

Script Spark akan menulis dua DataFrame hasil analisis menjadi dua tabel terpisah di dalam PostgreSQL:

1. market\_basket\_analysis
2. customer\_segmentation

Proses data writing dilakukan dengan mode overwrite. Ini memastikan bahwa setiap kali pipeline berjalan, tabel akan selalu berisi hasil analisis yang up to date, sehingga mencegah adanya data duplikat dari proses sebelumnya.

PostgreSQL menjadi single source of truth yang clean dan terstruktur. Nantinya Metabase akan terhubung langsung ke dua tabel ini untuk membuat dashboard.

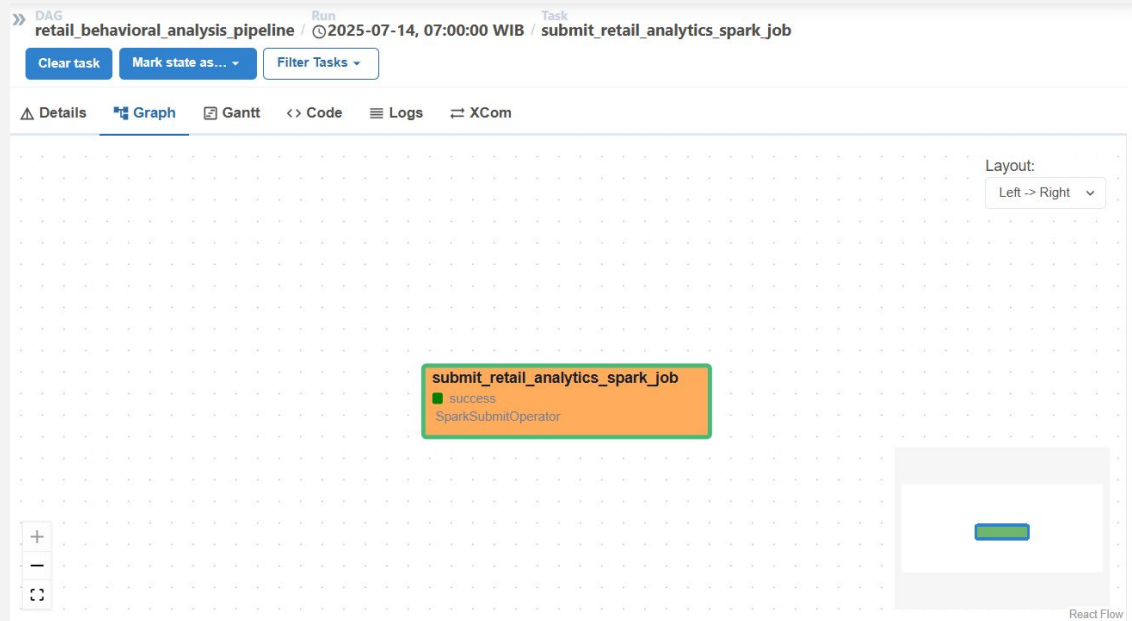
## Pipeline Orchestration with Airflow

Seluruh proses transformasi data yang telah dijelaskan akan diotomatisasi dan dijadwalkan daily menggunakan Apache Airflow.

Airflow bertanggung jawab untuk menjalankan pipeline secara berkala sesuai jadwal yang ditentukan yaitu @daily. Ini memastikan data di data mart selalu diperbarui dengan data transaksi terbaru.

Integrasi dengan Spark. Task ini diimplementasikan menggunakan SparkSubmitOperator dalam menjembatani Airflow dengan Spark, mengirimkan perintah untuk memulai proses ETL.

# Airflow UI



## DAG Code

```
# Define the DAG
with DAG(
    dag_id='retail_behavioral_analysis_pipeline',
    default_args=default_args,
    description='A DAG to run Spark job for Customer Segmentation (RFM) and Market Basket Analysis (MBA).',
    schedule_interval='@daily', # Run once every day
    catchup=False,
    tags=['retail', 'spark', 'behavioral-analysis'],
) as dag:

    # Define the Spark Submit task in local mode
    # This task will run Spark job locally without a cluster
    submit_spark_job = SparkSubmitOperator(
        task_id='submit_retail_analytics_spark_job',
        application='/opt/bitnami/spark/jobs/transform_job.py',
        verbose=False,
        # Use local mode configuration
        conf={'spark.master': 'local[*]'},
        # Use newer PostgreSQL driver version
        packages='org.postgresql:postgresql:42.7.3'
    )
```

A large black parallelogram on the left side of the slide, with two overlapping orange parallelograms positioned below it, creating a modern, abstract design.

# Data Modeling (Business)

# Data Modeling for Behavioral Analysis

Model data untuk platform ini dirancang untuk mendukung dua tujuan analisis yang berbeda, yaitu Market Basket Analysis dan Segmentasi Pelanggan. Oleh karena itu, data mart yang dibangun di PostgreSQL terdiri dari dua tabel agregat final yang masing-masing dioptimalkan untuk tujuannya.

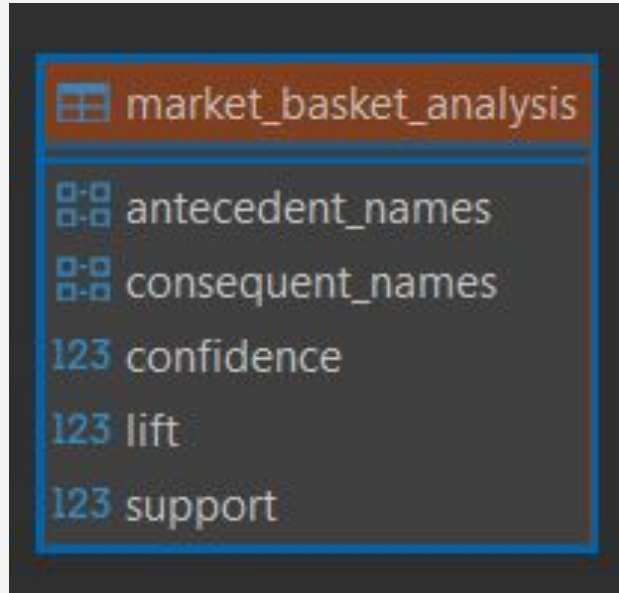
- **Tabel 1: market\_basket\_analysis**
  - Tujuan: Menyimpan hasil dari algoritma FP-Growth. Tabel ini berisi aturan asosiasi produk yang siap digunakan untuk keputusan strategis seperti penataan produk dan promosi bundling.
  - Struktur: antecedent\_names (produk awal), consequent\_names (produk yang berasosiasi), confidence, lift, dan support (metrik kekuatan asosiasi).

# Data Modeling for Behavioral Analysis

- **Tabel 2: customer\_segmentation**
  - Tujuan: Menyimpan hasil dari analisis RFM. Tabel ini memberikan profil untuk setiap pelanggan, memungkinkan perusahaan untuk merancang strategi marketing dan retensi yang tertarget.
  - Struktur: Customer ID, Recency, Frequency, Monetary, dan customer\_segment (label segmen seperti 'Champions', 'At Risk', dll).

Pendekatan dengan dua tabel terpisah ini memastikan bahwa query untuk setiap jenis analisis berjalan sangat cepat dan efisien, karena data sudah diagregasi dan disimpan sesuai dengan kebutuhan bisnisnya.

## Market Basket Analysis Table

A screenshot of a database table named 'market\_basket\_analysis'. The table has a dark background with a blue border. The header row is highlighted in orange and contains the table name. The data rows are listed below the header, each preceded by a small icon (a square with a dot) and a blue number '123'.

market_basket_analysis
antecedent_names
consequent_names
confidence
lift
support



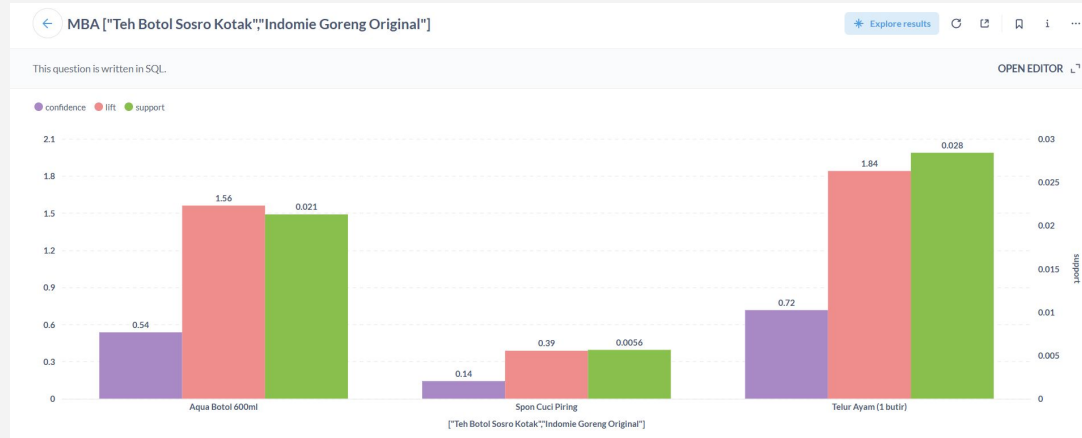
## RFM Table

customer_segmentation	
123	Customer ID
123	Recency
123	Frequency
123	Monetary
A-Z	customer_segment

A large black parallelogram on the left side of the slide, with two overlapping orange parallelograms positioned below it, creating a modern, abstract geometric design.

# Conclusion & Recommendation

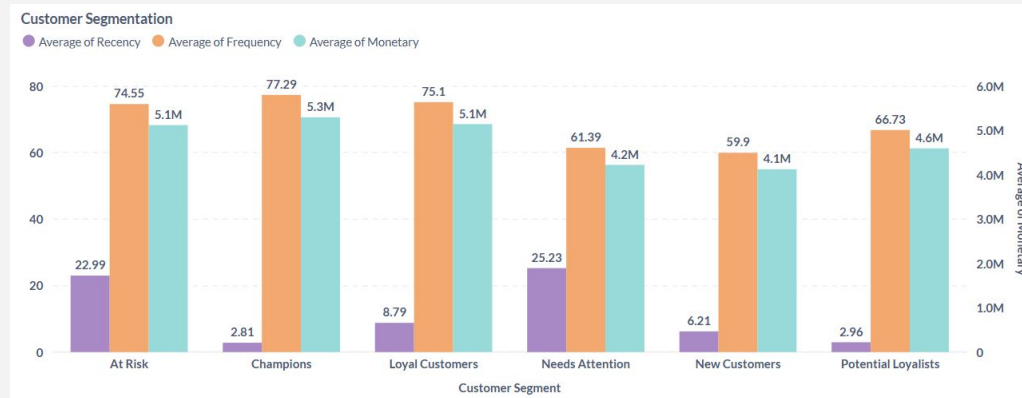
# Insight dari Market Basket Analysis



Dengan melihat visualisasi Market Basket Analysis, manajer toko dapat mengidentifikasi pola belanja yang lebih terarah dan actionable.

Sebagai contoh, hasil analisis menunjukkan bahwa ketika pelanggan membeli "Teh Botol Sosro Kotak" dan "Indomie Goreng Original" secara bersamaan, mereka sangat mungkin juga membeli "Telur Ayam (1 butir)", menunjukkan pola belanja bahan makanan instan yang saling melengkapi.

# Insight dari Segmentasi Pelanggan (RFM)



Segmen "Champions" adalah pelanggan paling berharga. Mereka memiliki nilai Recency yang sangat rendah, artinya mereka baru saja melakukan pembelian. Selain itu, mereka juga memiliki Frequency dan Monetary yang sangat tinggi, menunjukkan bahwa mereka sering membeli dan mengeluarkan banyak uang dalam setiap transaksi.

# Insight dari Segmentasi Pelanggan (RFM)



Pelanggan seperti ini adalah aset penting yang harus dijaga dengan baik. Tim marketing bisa memberikan loyalty program eksklusif atau reward personal sebagai bentuk apresiasi untuk menjaga hubungan jangka panjang dan mendorong mereka menjadi brand advocate.

# Metabase Dashboard



## Conclusion

Melalui Market Basket Analysis, ditemukan pola pembelian produk yang sering dibeli bersamaan, seperti kombinasi Teh Botol Sosro Kotak, Indomie Goreng Original, dan Telur Ayam (1 butir), yang mencerminkan preferensi konsumen terhadap makanan instan siap saji. Pola ini memberi peluang besar untuk strategi bundling dan penataan produk yang lebih tepat sasaran di toko.

Di sisi lain, segmentasi RFM mengidentifikasi kelompok Champions yaitu pelanggan yang baru saja berbelanja dengan frekuensi tinggi dan pengeluaran besar. Mereka adalah aset penting bagi bisnis dan sangat potensial untuk dijadikan target program loyalitas atau reward eksklusif agar hubungan jangka panjang tetap terjaga.

Dengan menggabungkan kedua analisis ini, keputusan pemasaran menjadi lebih terarah dan berbasis data, bukan sekadar intuisi.

A large, stylized graphic on the left side of the slide. It consists of a blue outline of a person's head and shoulders. Inside the head is a series of concentric circles: a small light orange circle, a medium orange circle, and a large blue circle. The body is a large blue circle with a large orange circle in the center, which also contains a smaller light orange circle.

**Terima  
Kasih.**