## 

## 

## Cover Page

## 

**User Manual**

Dokumen ini bertujuan untuk mempermudah pembaca dalam memahami rancangan dan struktur database

## 

## 

## Table of Contents

[**Cover Page 1**](#_4qccuwvtdfz8)

[**Table of Contents 2**](#_tsnnnm46dg1b)

[**Entity Relationship Diagram 3**](#_usf7ldps9sqi)

[**Data Catalog 4**](#_adbwepnglum6)

[Table Customer 4](#_cjb8yq11jxjb)

[Table Order 4](#_jzgef65rb4eq)

[Table Resto 4](#_qc34n066jz3s)

[Table Food 5](#_98elz6vtz7bi)

[Table Driver 5](#_m1v53w9bmpgb)

[**Report Catalog 6**](#_l9wjzmkmd43z)

[Table Daily Registrasi 6](#_jtv5lbrdosbk)

[Table Jumlah Order Harian 6](#_a51a4a8bsdpq)

[Table Keuntungan Harian 7](#_p9jzc1vjzlup)

[Table Average spending customer per Month 7](#_bvq6x05rpx1s)

[Table Segmentasi Customer 8](#_kfx48engkfcl)

[Table Order yang berstatus Pending 8](#_kgg6qsv2dtf)

[Table Revenue dalam 30 Hari 10](#_dd0l7f5564hy)

[Table Total Revenue per Hari dalam 30 Hari 10](#_g1xjbhe0vd5q)

[Table Total Revenue per Hari per Kota dalam 30 Hari 11](#_ukwoa84jabwj)

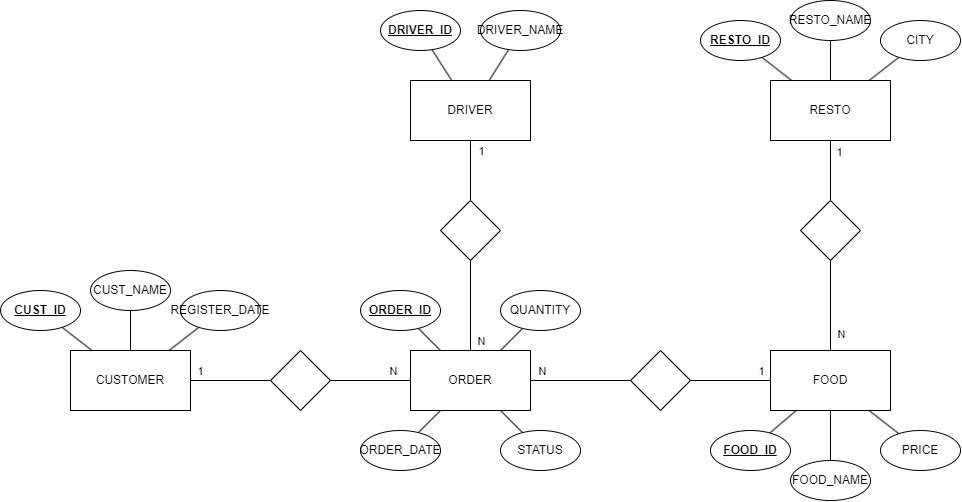
[Table Segmentasi Customer dalam 30 hari 12](#_6yvb2120oimm)

[Table Kontribusi Kota per Hari dalam 30 Hari 14](#_16sad3uhalf2)

[Table Growth Harian per Kota dalam 30 Hari 16](#_ogwr95tari8h)

## 

## Entity Relationship Diagram



| *List of Table* | *Description* | *Limitation* | *Notes* |
| --- | --- | --- | --- |
| Customer | Menyimpan data personal seluruh pengguna aplikasi OKFoodDelivery meliputi id, nama, dan tanggal registrasi | Hanya menyimpan id, nama dan tanggal registrasi | Tidak memiliki primary key / foreign key |
| Order | Menyimpan data transaksi seluruh pengguna aplikasi OkFoodDelivery | Menyimpan id, quantity, status pesanan, tanggal pemesanan | Memiliki foreign key dari table customer, table food, table driver |
| Food | Menyimpan data seluruh menu makanan dari restorant yang listing pada aplikasi OkFoodDelivery | Menyimpan id, nama makanan, harga makanan | Memiliki foreign key dari table resto |
| Resto | Menyimpan data seluruh restoran yang listing di aplikasi OkFoodDelivery | Menyimpan id, nama resto, dan kota resto tersebut | Tidak memiliki foreign key |
| Driver | Menyimpan data personal driver | Menyimpan id dan nama | Tidak memiliki foreign key |

## Data Catalog

### Table Customer

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| cust\_id | Integer | Id dari pengguna | Hanya terdiri dari angka saja |
| cust\_name | String | Nama dari pengguna | Hanya terdiri dari huruf saja |
| register\_date | Date | Tanggal registrasi pengguna | Hanya terdiri dari tanggal/ date saja |

### Table Order

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| order\_id | Integer | Id dari pesanan / order | Hanya terdiri dari angka saja |
| order\_date | Date | Tanggal pemesanan | Hanya terdiri dari tanggal saja |
| quantity | Integer | Banyaknya pesanan | Hanya terdiri dari angka saja |
| status | String | Status pesanan | Hanya terdiri dari huruf saja |
| fk\_cust\_id | Integer | Id dari pengguna | Hanya terdiri dari angka saja |
| fk\_food\_id | Integer | Id makanan dari resto yang listing di aplikasi | Hanya terdiri dari angka saja |
| fk\_driver\_id | Integer | Id dari driver | Hanya terdiri dari angka saja |

### Table Resto

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| resto\_id | Integer | Id dari resto | Hanya terdiri dari angka saja |
| resto\_name | String | Nama dari resto | Hanya terdiri dari huruf saja |
| city | String | Kota dari resto | Hanya terdiri dari huruf |

### Table Food

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| food\_id | Integer | Id dari makanan | Hanya terdiri dari angka saja |
| food\_name | String | Nama dari menu / makanan | Hanya terdiri dari huruf saja |
| price | Integer | Harga dari makanan | Hanya terdiri dari angka saja |
| fk\_resto\_id | Integer | Id dari resto | Hanya terdiri dari angka saja |

### Table Driver

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| driver\_id | Integer | Id dari driver | Hanya terdiri dari angka saja |
| driver\_name | String | Nama dari driver | Hanya terdiri dari huruf saja |

## 

## 

## Report Catalog

// berisikin semua report yang teman-teman buat / tabel agregasi

### Table Daily Registrasi

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| register\_date | Date | Tanggal pengguna bergabung di aplikasi / registrasi | Hanya terdiri dari tanggal saja |
| jumlah\_customer | Integer | Jumlah customer | Hanya terdiri dari angka |

#### Query

| SELECT register\_date, COUNT(cust\_id) AS Jumlah\_order  FROM `aqueous-freedom-398208.dkf\_okefood.customer`  GROUP BY register\_date  ORDER BY register\_date ASC; |
| --- |

### Table Jumlah Order Harian

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| register\_date | Date | Tanggal pengguna bergabung di aplikasi / registrasi | Hanya terdiri dari tanggal saja |
| jumlah\_order | Integer | Jumlah order | Hanya terdiri dari angka |

#### Query

| SELECT register\_date, COUNT(cust\_id) AS Jumlah\_order  FROM `aqueous-freedom-398208.dkf\_okefood.customer`  GROUP BY register\_date  ORDER BY register\_date ASC; |
| --- |

### Table Keuntungan Harian

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| order\_date | Date | Tanggal pengguna melakukan pesanan | Hanya terdiri dari tanggal saja |
| Total\_Keuntungan | Integer | Total keuntungan | Hanya terdiri dari angka |

#### Query

| SELECT  order\_date,  SUM(price\*quantity) AS Total\_Keuntungan  From `aqueous-freedom-398208.dkf\_okefood.order` O  JOIN `aqueous-freedom-398208.dkf\_okefood.food` F  ON O.fk\_food\_id = F.food\_id  GROUP BY order\_date  ORDER BY order\_date ASC |
| --- |

### Table Average spending 1 customer per Month

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| Cust\_name | Varchar | Nama customer | Hanya terdiri dari huruf |
| Month\_ | Date | Tanggal | Hanya terdiri dari tanggal saja |
| Average\_daily\_spend | Float | Rata-rata dari total keuntungan | Hanya terdiri dari angka |

#### Query

| SELECT  cust\_name,  DATE\_TRUNC(order\_date, MONTH) as Month\_,  ROUND(AVG( Total\_Keuntungan)) AS Average\_daily\_spend  FROM (SELECT cust\_name, order\_date, SUM(price\*quantity) AS Total\_Keuntungan  From `aqueous-freedom-398208.dkf\_okefood.order` O  JOIN `aqueous-freedom-398208.dkf\_okefood.food` F ON O.fk\_food\_id = F.food\_id  JOIN `aqueous-freedom-398208.dkf\_okefood.customer` C ON O.fk\_cust\_id = C.cust\_id  GROUP BY order\_date, cust\_name  )  GROUP BY 1,2  ORDER BY 2 ASC; |
| --- |

### Table Customer Null

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| null\_cust\_id | Integer | Mencari null di kolom cust\_id | Count null di kolom cust\_id |
| null\_cust\_name | integer | Mencari null di kolom cust\_name | Count null di kolom cust\_name |
| null\_register\_date | Integer | Mencari null di kolom register\_date | Count null di kolom register\_date |

#### Query

| SELECT COUNT (cust\_id) AS null\_cust\_id, COUNT(cust\_name) AS null\_cust\_name, COUNT(register\_date)AS null\_register\_date  FROM `aqueous-freedom-398208.dkf\_okefood.customer`  WHERE cust\_id IS NULL OR cust\_name IS NULL OR register\_date IS NULL; |
| --- |

### 

### Table food Null

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| null\_food\_id | Integer | Mencari null di kolom food\_id | Count null di kolom food\_id |
| null\_food\_name | integer | Mencari null di kolom food\_name | Count null di kolom food\_name |
| null\_price | Integer | Mencari null di kolom price | Count null di kolom price |

#### Query

| SELECT COUNT(food\_id) AS null\_food\_id , COUNT(food\_name) AS null\_food\_name, COUNT(price) AS null\_price  FROM `aqueous-freedom-398208.dkf\_okefood.food`  WHERE food\_id IS NULL OR food\_name IS NULL OR price IS NULL |
| --- |

### 

### Table order Null

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| null\_order\_id | Integer | Mencari null di kolom order\_id | Count null di kolom order\_id |
| null\_order\_date | Integer | Mencari null di kolom order\_date | Count null di kolom order\_date |
| null\_quantity | Integer | Mencari null di kolom quantity | Count null di kolom quantity |
| null\_status | Integer | Mencari null di kolom status | Count null di kolom status |

#### Query

| SELECT COUNT(order\_id) AS null\_order\_id, COUNT(order\_date) AS null\_order\_date, COUNT(quantity) AS null\_quantity, COUNT(status) AS null\_status  FROM `aqueous-freedom-398208.dkf\_okefood.order`  WHERE order\_id IS NULL OR order\_date IS NULL OR quantity IS NULL OR status IS NULL |
| --- |

### 

### Table resto Null

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| null\_resto\_id | Integer | Mencari null di kolom resto\_id | Count null di kolom resto\_id |
| null\_resto\_name | integer | Mencari null di kolom resto\_name | Count null di kolom resto\_name |
| null\_city | Integer | Mencari null di kolom city | Count null di kolom city |

#### Query

| SELECT COUNT(resto\_id) AS null\_resto\_id , COUNT(resto\_name) AS null\_resto\_name,COUNT(city) AS null\_city  FROM `aqueous-freedom-398208.dkf\_okefood.resto`  WHERE resto\_id IS NULL OR resto\_name IS NULL OR city IS NULL |
| --- |

### 

### Table driver Null

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| null\_driver\_id | Integer | Mencari null di kolom driver\_id | Count null di kolom driver\_id |
| null\_driver\_name | integer | Mencari null di kolom driver\_name | Count null di kolom driver\_name |

#### Query

| SELECT COUNT(driver\_id) AS null\_driver\_id , COUNT(driver\_name) AS null\_driver\_name  FROM `aqueous-freedom-398208.dkf\_okefood.driver`  WHERE driver\_id IS NULL OR driver\_name IS NULL |
| --- |

### 

### Table Segmentasi Customer

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| cust\_name | String | Nama customer | Antara cust\_name dan name merujuk pada fungsi kolom yang sama |
| total\_order | Integer | Total jumlah order | Dihasilkan dari COUNT(\*) |
| total\_spend | Integer | Total uang yang dikeluarkan customer | Dihasilkan dari SUM(quantity\*price) |
| level | String | Kategori customer | Berdasarkan total\_order dan total\_spend |

#### Query

| SELECT sub.cust\_name, sub.total\_order, sub.total\_spend, CASE  WHEN sub.total\_order > 3 AND sub.total\_spend >= 80000 THEN 'Big Spender'  WHEN sub.total\_spend < 10000 THEN 'Regular'  WHEN sub.total\_spend < 40000 THEN 'Silver'  WHEN sub.total\_spend < 80000 THEN 'Gold'  ELSE 'Diamond'  END level  FROM (  SELECT c.cust\_name, COUNT(\*) AS total\_order, SUM(f.price\*o.quantity) AS total\_spend  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  GROUP BY 1  ) AS sub  ORDER BY 3, 4; |
| --- |

### Table Order yang berstatus Pending

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| cust\_name | String | Nama customer | Antara cust\_name dan name merujuk pada fungsi kolom yang sama |
| order\_date | Date | Tanggal order | Antara order\_date dan transaction\_date merujuk pada fungsi kolom yang sama |
| status | Sting | Status order | Bernilai ‘Pending’, ‘Diantar’, dan ‘Selesai’ |

#### Query

| SELECT cust\_name, order\_date, status  FROM `aqueous-freedom-398208.dkf\_okefood.customer` C  JOIN `aqueous-freedom-398208.dkf\_okefood.order` O ON C.cust\_id = O.fk\_cust\_id  WHERE status = 'Pending'  ORDER BY order\_date ASC; |
| --- |

### 

### Table Revenue dalam 30 Hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| name | String | Nama customer | Antara cust\_name dan name merujuk pada fungsi kolom yang sama |
| kota | String | Kota dari resto | Antara city dan kota merujuk pada fungsi kolom yang sama |
| transaction\_date | Date | Tanggal order | Antara order\_date dan transaction\_date merujuk pada fungsi kolom yang sama |
| price | Integer | Harga food |  |
| quantity | Integer | Jumlah food yang dipesan |  |

#### Query Dataset Tim 2

| SELECT c.cust\_name, r.city, o.order\_date, f.price, o.quantity  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  WHERE o.order\_date > CURRENT\_DATE()-30 AND o.order\_date <= CURRENT\_DATE() |
| --- |

#### Query Dataset Coding.ID

| SELECT c.name, r.kota, t.transaction\_date, t.price  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE() |
| --- |

### Table Total Revenue per Hari dalam 30 Hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| transaction\_date | Date | Tanggal order | Antara order\_date dan transaction\_date merujuk pada fungsi kolom yang sama |
| total\_revenue | Integer | Total revenue per hari | Dihasilkan dari SUM(quantity\*price) |

#### Query Dataset Tim 2

| SELECT DATE(o.order\_date)AS transaction\_date, SUM(f.price\*o.quantity) AS total\_revenue  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  WHERE o.order\_date > CURRENT\_DATE()-30 AND o.order\_date <= CURRENT\_DATE()  GROUP BY 1  ORDER BY 1 |
| --- |

#### Query Dataset Coding.ID

| SELECT t.transaction\_date, SUM(t.price) AS total\_revenue,  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE()  GROUP BY 1 |
| --- |

### Table Total Revenue per Hari per Kota dalam 30 Hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| transaction\_date | Date | Tanggal order | Antara order\_date dan transaction\_date merujuk pada fungsi kolom yang sama |
| city | String | Kota dari resto | Antara city dan kota merujuk pada fungsi kolom yang sama |
| city\_revenue | Date | Total revenue per hari per kota | Dihasilkan dari SUM(quantity\*price) |

#### Query Dataset Tim 2

| SELECT DATE(o.order\_date) AS transaction\_date, r.city, SUM(o.quantity \* f.price) AS city\_revenue  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  GROUP BY DATE(o.order\_date), r.city |
| --- |

#### Query Dataset Coding.ID

| SELECT r.kota, t.transaction\_date, SUM(t.price) AS city\_revenue,  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE()  GROUP BY 1, 2 |
| --- |

### Table Segmentasi Customer dalam 30 hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| name | String | Tanggal order |  |
| total\_transaction | Integer | Total jumlah transaksi order per customer | Dihasilkan dari COUNT(\*) |
| total\_spending | Integer | Total uang yang dikeluarkan per customer | Dihasilkan dari SUM(price\*quantity) |
| segmentation | String | Kategori dari customer | Berdasarkan total\_spending dan total\_transaction  Data yang ada pada dataset tim 2 kurang bervariasi sehingga kurang memperlihatkan kategorisasi customer |

#### Query Dataset Tim 2

| WITH sub AS (  SELECT sub2.cust\_name, COUNT(\*) AS total\_transaction, SUM(sub2.price\*sub2.quantity) AS total\_spending  FROM (  SELECT c.cust\_name, o.order\_date, f.price, o.quantity  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  WHERE o.order\_date > CURRENT\_DATE()-30 AND o.order\_date <= CURRENT\_DATE()  ) AS sub2  GROUP BY 1  )  SELECT sub.name, sub.total\_transaction, sub.total\_spending,  CASE  WHEN sub.total\_spending > 500000 THEN 'Big Spender'  WHEN sub.total\_transaction < 2 THEN 'New'  WHEN sub.total\_transaction < 6 THEN 'Good'  WHEN sub.total\_transaction < 11 THEN 'Potential'  ELSE 'Loyalist'  END segmentation  FROM sub  ORDER BY 1; |
| --- |

#### Query Dataset Coding.ID

| WITH sub AS (  SELECT sub2.name, COUNT(\*) AS total\_transaction, SUM(sub2.price) AS total\_spending  FROM (  SELECT c.name, t.transaction\_date, t.price  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE()  ) AS sub2  GROUP BY 1  )  SELECT sub.name, sub.total\_transaction, sub.total\_spending,  CASE  WHEN sub.total\_spending > 500000 THEN 'Big Spender'  WHEN sub.total\_transaction < 2 THEN 'New'  WHEN sub.total\_transaction < 6 THEN 'Good'  WHEN sub.total\_transaction < 11 THEN 'Potential'  ELSE 'Loyalist'  END segmentation  FROM sub  ORDER BY 1; |
| --- |

### Table Kontribusi Kota per Hari dalam 30 Hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| transaction\_date | Date | Tanggal order |  |
| kota | String | Kota dari restoran |  |
| city\_revenue | Integer | Total revenue sebuah kota | Dihasilkan dari SUM(quantity\*price) GROUP BY transaction\_date dan kota |
| total\_revenue | Integer | Total revenue semua kota pada tanggal tersebut atau pada satu hari | Dihasilkan dari SUM(quantity\*price)  GROUP BY transaction\_date |
| contribution\_percent | Float | Persentase kontribusi sebuah kota | Dihasilkan dari city\_revenue/daily\_revenue |

#### Query Dataset Tim 2

| WITH daily\_revenue AS (  SELECT DATE(o.order\_date) AS transaction\_date, r.city, SUM(o.quantity \* f.price) AS city\_revenue  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  WHERE o.order\_date > CURRENT\_DATE()-30 AND o.order\_date <= CURRENT\_DATE()  GROUP BY DATE(o.order\_date), r.city  ),  total\_daily\_revenue AS (  SELECT DATE(o.order\_date)AS transaction\_date, SUM(f.price\*o.quantity) AS total\_revenue,  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.customer` c ON c.cust\_id = o.fk\_cust\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  WHERE o.order\_date > CURRENT\_DATE()-30 AND o.order\_date <= CURRENT\_DATE()  GROUP BY 1  ORDER BY 1  )  SELECT dr.transaction\_date, dr.city AS kota, dr.city\_revenue, tdr.total\_revenue,  ROUND((dr.city\_revenue / tdr.total\_revenue) \* 100) AS contribution\_percent  FROM daily\_revenue dr  JOIN total\_daily\_revenue tdr ON dr.transaction\_date = tdr.transaction\_date  ORDER BY dr.transaction\_date, dr.city; |
| --- |

#### Query Dataset Coding.ID

| WITH daily\_revenue AS (  SELECT r.kota, t.transaction\_date, SUM(t.price) AS city\_revenue,  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE()  GROUP BY 1, 2  ),  total\_daily\_revenue AS (  SELECT t.transaction\_date, SUM(t.price) AS total\_revenue,  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.customer` c ON c.id = t.customer\_id  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  WHERE t.transaction\_date > CURRENT\_DATE()-30 AND t.transaction\_date <= CURRENT\_DATE()  GROUP BY 1  )  SELECT dr.transaction\_date, dr.kota, dr.city\_revenue, tdr.total\_revenue, ROUND((dr.city\_revenue / tdr.total\_revenue) \* 100) AS contribution\_percent  FROM daily\_revenue dr  JOIN total\_daily\_revenue tdr ON dr.transaction\_date = tdr.transaction\_date  ORDER BY dr.transaction\_date, dr.kota; |
| --- |

### Table Growth Harian per Kota dalam 30 Hari

#### Table Structure

| *Column Name* | *Data Type* | *Description* | *Notes* |
| --- | --- | --- | --- |
| kota | String | Kota dari resto |  |
| transaction\_date | Date | Tanggal transaksi |  |
| daily\_revenue | Integer | Total revenue per hari | Dihasilkan dari SUM(quantity\*price) |
| growth | Float | Pertumbuhan revenue dari hari sebelumnya | Dihasilkan dari (total revenue - total revenue d-1) / total revenue d-1  Untuk menghasilkan data growth yang mudah dipahami sangat tergantung pada urutan data. Jika urutan tidak sesuai maka data sulit untuk dipahami |

#### Query Dataset Tim 2

| INSERT INTO `aqueous-freedom-398208.dkf\_summary\_tim2.summary\_growth`  WITH sub AS (  SELECT r.city AS kota, DATE(o.order\_date) AS transaction\_date, COALESCE(SUM(f.price \* o.quantity), NULL) AS daily\_revenue  FROM `aqueous-freedom-398208.dkf\_okefood.order` o  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.food` f ON f.food\_id = o.fk\_food\_id  INNER JOIN `aqueous-freedom-398208.dkf\_okefood.resto` r ON r.resto\_id = f.fk\_resto\_id  GROUP BY 1, 2  ),  currents AS (  SELECT DISTINCT kota, CURRENT\_DATE() AS transaction\_date  FROM sub  ),  result AS (  SELECT s.kota, s.transaction\_date, COALESCE(sub.daily\_revenue, NULL) AS daily\_revenue  FROM currents s  LEFT JOIN sub ON s.kota = sub.kota AND s.transaction\_date = sub.transaction\_date  ),  compare AS (  SELECT r.kota, r.transaction\_date, r.daily\_revenue  FROM result r  UNION ALL (SELECT kota, transaction\_date, daily\_revenue FROM `aqueous-freedom-398208.dkf\_summary\_tim2.summary\_growth`)  ),  fill AS (  SELECT c.kota, c.transaction\_date, COALESCE(c.daily\_revenue, LAG(c.daily\_revenue) OVER(PARTITION BY c.kota ORDER BY c.transaction\_date)) AS daily\_revenue  FROM compare c  ),  growth AS (  SELECT f.kota, f.transaction\_date, f.daily\_revenue, ROUND((f.daily\_revenue - LAG(f.daily\_revenue) OVER(PARTITION BY f.kota ORDER BY f.transaction\_date)) / NULLIF(LAG(f.daily\_revenue) OVER(PARTITION BY f.kota ORDER BY f.transaction\_date), 0) \* 100, 2) AS growth  FROM fill f  )  SELECT g.kota, g.transaction\_date, g.daily\_revenue, g.growth  FROM growth g  WHERE g.transaction\_date = CURRENT\_DATE()  ORDER BY 1,2 |
| --- |

#### Query Dataset Coding.ID

| INSERT INTO `aqueous-freedom-398208.dkf\_summary\_codingid.summary\_growth`  WITH sub AS (  SELECT r.kota, DATE(t.transaction\_date) AS transaction\_date, COALESCE(SUM(t.price), NULL) AS daily\_revenue  FROM `aqueous-freedom-398208.ok\_food\_dataset.transaction\_2024` t  INNER JOIN `aqueous-freedom-398208.ok\_food\_dataset.resto` r ON r.id = t.resto\_id  GROUP BY 1, 2  ),  currents AS (  SELECT DISTINCT kota, CURRENT\_DATE() AS transaction\_date  FROM sub  ),  result AS (  SELECT s.kota, s.transaction\_date, COALESCE(sub.daily\_revenue, NULL) AS daily\_revenue  FROM currents s  LEFT JOIN sub ON s.kota = sub.kota AND s.transaction\_date = sub.transaction\_date  ),  compare AS (  SELECT r.kota, r.transaction\_date, r.daily\_revenue  FROM result r  UNION ALL (SELECT kota, transaction\_date, daily\_revenue  FROM `aqueous-freedom-398208.dkf\_summary\_codingid.summary\_growth`)  ),  fill AS (  SELECT c.kota, c.transaction\_date, COALESCE(c.daily\_revenue, LAG(c.daily\_revenue) OVER(PARTITION BY c.kota ORDER BY c.transaction\_date)) AS daily\_revenue  FROM compare c  ),  growth AS (  SELECT f.kota, f.transaction\_date, f.daily\_revenue, ROUND((f.daily\_revenue - LAG(f.daily\_revenue) OVER(PARTITION BY f.kota ORDER BY f.transaction\_date)) / NULLIF(LAG(f.daily\_revenue) OVER(PARTITION BY f.kota ORDER BY f.transaction\_date), 0) \* 100, 2) AS growth  FROM fill f  )  SELECT g.kota, g.transaction\_date, COALESCE(g.daily\_revenue, 0), g.growth  FROM growth g  WHERE g.transaction\_date = CURRENT\_DATE()  ORDER BY 1,2 |
| --- |

#### 