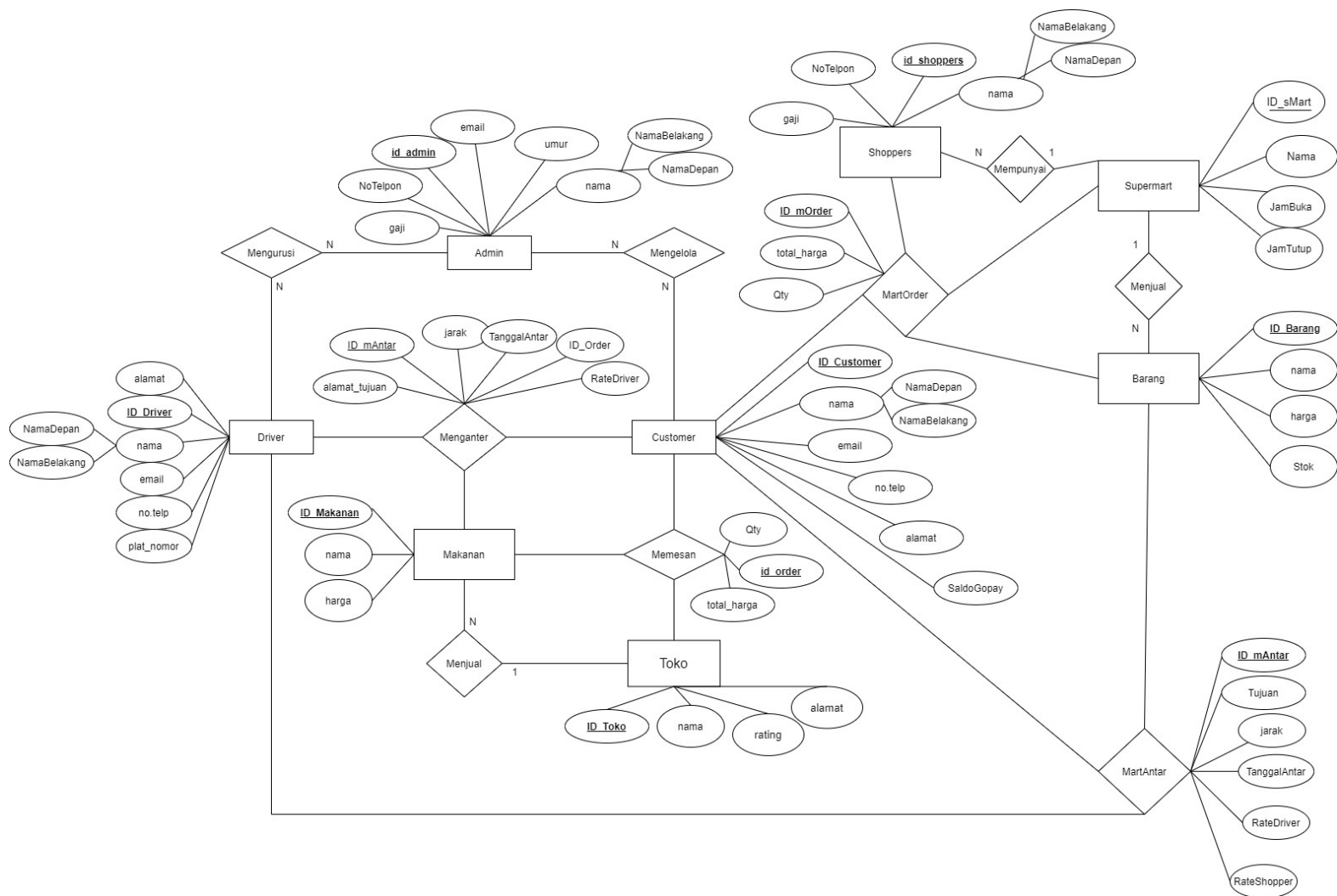
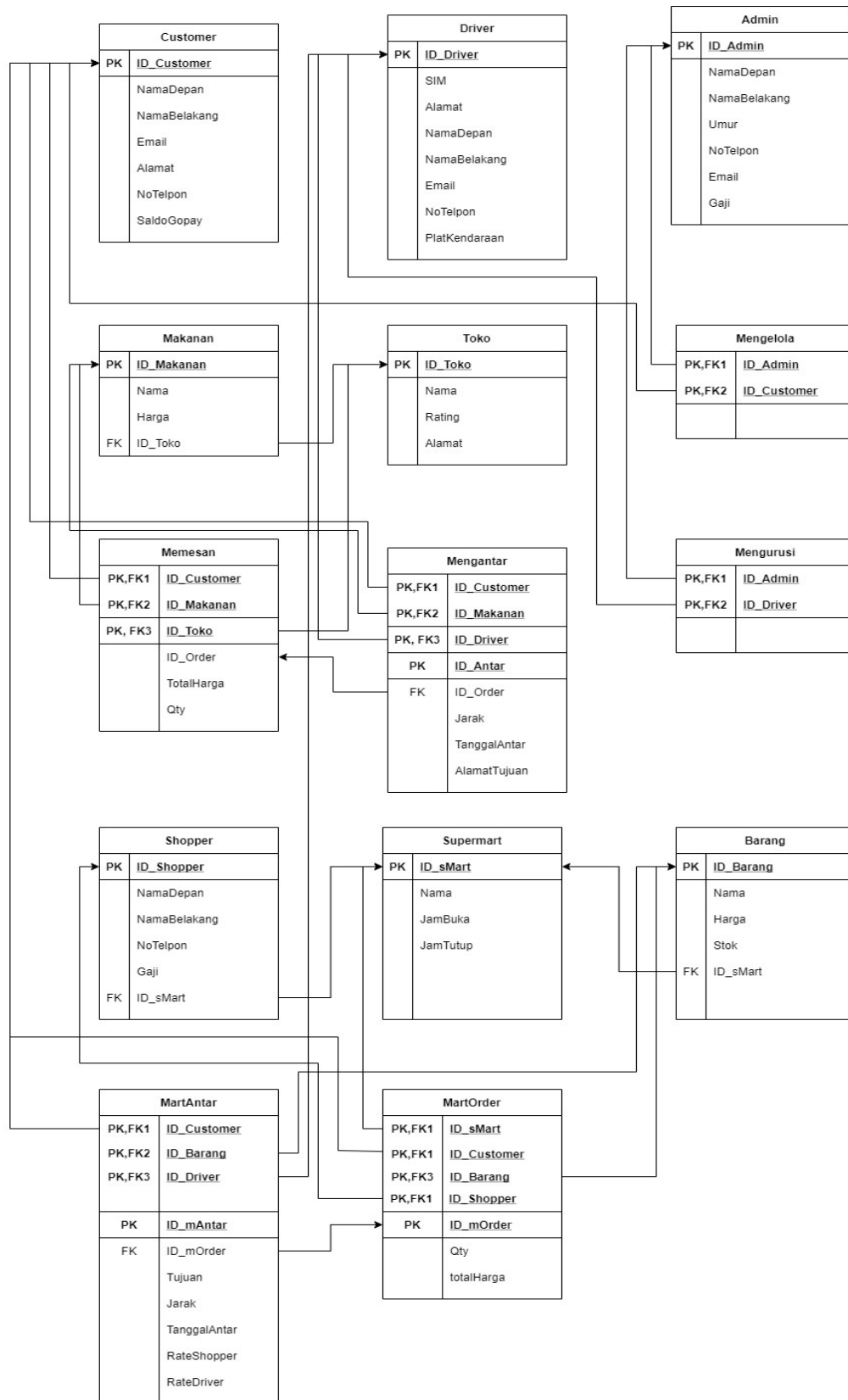


ERD



Database Schema



1. Statement

a. Query pertama

- Fungsi: menampilkan nama customer yang pernah memesan di tanggal dan bulan yang sama, serta nama makanan yang dipesannya.
- ```
SELECT CONCAT(c>NamaDepan, " ",IFNULL(c>NamaBelakang,""))
Customer,
GROUP_CONCAT(m.TanggalAntar) Tanggal,
GROUP_CONCAT(ma.nama) "Nama Makanan" FROM Customer c JOIN
Mengantar m USING (ID_Customer) JOIN Makanan ma USING
(ID_Makanan) WHERE TanggalAntar IN (SELECT TanggalAntar FROM
Mengantar WHERE MONTH(TanggalAntar)= DAY(TanggalAntar))
GROUP BY 1;
```
- Hasil eksekusi:

| Customer        | Tanggal               | Nama Makanan                    |
|-----------------|-----------------------|---------------------------------|
| Agus Setyo      | 2021-05-05            | Nasi Uduk                       |
| Atras Chandra   | 2021-08-08,2021-06-06 | Tempe Orek,Pizza Jamur          |
| Dwi Widya       | 2021-09-09            | Kerupuk Kulit                   |
| Farida Setiawan | 2021-03-03,2021-11-11 | Nasi Goreng Special,Telur Pedas |
| Nisa Ayu        | 2021-05-05            | EsKrim Coklat                   |
| Rafa Hermansyah | 2021-07-07,2021-04-04 | Kue Hantu,Pizza Aja             |
| Risa Adinda     | 2021-01-01            | Nasi Uduk                       |
| Salim Putra     | 2021-12-12            | Sate Ayam                       |

### b. Query kedua

- Fungsi: menampilkan seluruh nama driver beserta bonus tip yang diterima berdasarkan jarak yang ditempuh
- ```
SELECT CONCAT(d>NamaDepan, " ",IFNULL(d>NamaBelakang,""))
Driver,
CASE
    WHEN SUM(m.Jarak) IS NULL THEN 'Semangat Mencari Nafkah'
    WHEN SUM(m.Jarak) < 10 THEN CONCAT("Rp.
",SUM(m.Jarak)*10000)
    WHEN SUM(m.Jarak) < 15 THEN CONCAT("Rp.
",SUM(m.Jarak)*15000)
    WHEN SUM(m.Jarak) < 25 THEN CONCAT("Rp.
",SUM(m.Jarak)*20000)
    WHEN SUM(m.Jarak) >= 25 THEN CONCAT("Rp.
",SUM(m.Jarak)*250000)
END Bonus
FROM Driver d LEFT JOIN Mengantar m USING (ID_Driver) GROUP BY
1 ORDER BY SUM(m.Jarak) DESC;
```

- Hasil eksekusi:

Driver	Bonus
Dian Cahaya	Rp. 7750000
Eko Putra	Rp. 7250000
Sinta Sari	Rp. 340000
Hermanto Syah	Rp. 210000
Ratna Devi	Rp. 172500
Indah Bulan	Rp. 165000
Slamet Darma	Rp. 150000
Susilo Irwan	Rp. 80000
Ade Irwan	Rp. 35000
Rahmat Basuki	Rp. 15000
Kino Andhika	Semangat Mencari Nafkah

c. Query ketiga

- Fungsi: menampilkan informasi admin
- SET @M = "Mengurusi ";
SET @Me = "Mengelola ";
SET @DR = " Akun Driver ";
SET @CS = " Akun Customer ";
SET @B = " yang berasal dari Jakarta";
SET @C = " yang berasal dari Cianjur";
SET @D = " yang berasal dari Bandung";
SELECT "Admin Yang Mengurusi Akun Driver" AS "Admin Information"
UNION
SELECT CONCAT(RPAD(CONCAT(a>NamaDepan, " ",
IFNULL(a>NamaBelakang,"")),17, " "),
 RPAD(CASE WHEN SUBSTRING(d.PlatKendaraan, 1, 1) = "B"
THEN CONCAT(@M, COUNT(*), @DR,
GROUP_CONCAT(d>NamaDepan SEPARATOR " dan "), @B)
 WHEN SUBSTRING(d.PlatKendaraan, 1, 1) = "C" THEN
CONCAT(@M, COUNT(*), @DR, GROUP_CONCAT(d>NamaDepan
SEPARATOR " dan "), @C)
 WHEN SUBSTRING(d.PlatKendaraan, 1, 1) = "D" THEN
CONCAT(@M, COUNT(*), @DR, GROUP_CONCAT(d>NamaDepan
SEPARATOR " dan "), @D)
 END,70," ")) FROM Mengurusi m JOIN Admin a USING
(ID_Admin) JOIN Driver d ON (d.ID_Driver = m.ID_Driver) GROUP BY
CONCAT(a>NamaDepan, " ", IFNULL(a>NamaBelakang,""))
UNION
SELECT " "
UNION
SELECT "Admin Yang Mengelola Akun Customer"
UNION

```

SELECT      CONCAT(RPAD(CONCAT(a>NamaDepan,      "      ",
IFNULL(a>NamaBelakang,"")),17, " "),
              RPAD(CONCAT(@Me,   COUNT(*),   @CS,
GROUP_CONCAT(c>NamaDepan SEPARATOR " dan ")),70," ")) FROM
Mengelola me JOIN Admin a USING (ID_Admin) JOIN Customer c ON
(c.ID_Customer      =      me.ID_Customer)      GROUP      BY
CONCAT(a>NamaDepan, " ", IFNULL(a>NamaBelakang,""));
SET @M = NULL;
SET @Me = NULL;
SET @DR = NULL;
SET @CS = NULL;
SET @B = NULL;
SET @C = NULL;
SET @D = NULL;

```

- Hasil eksekusi:

Admin Information		

Admin Yang Mengurusi Akun Driver		
Alien	Mengurusi 2 Akun Driver	Slamet dan Hermanto yang berasal dari Jakarta
Amel Lia	Mengurusi 2 Akun Driver	Ade dan Susilo yang berasal dari Cianjur
Christine	Mengurusi 2 Akun Driver	Hermanto dan Sinta yang berasal dari Bandung
Ela Setio	Mengurusi 2 Akun Driver	Dian dan Ratna yang berasal dari Bandung
Groot	Mengurusi 2 Akun Driver	Indah dan Ratna yang berasal dari Jakarta
Jojo Nathan	Mengurusi 2 Akun Driver	Eko dan Ade yang berasal dari Jakarta
Klaudi Banu	Mengurusi 2 Akun Driver	Susilo dan Rahmat yang berasal dari Cianjur
Michael Alex	Mengurusi 2 Akun Driver	Sinta dan Rahmat yang berasal dari Jakarta
Niko Adri	Mengurusi 2 Akun Driver	Eko dan Slamet yang berasal dari Jakarta
Wendy Anto	Mengurusi 2 Akun Driver	Indah dan Dian yang berasal dari Jakarta
Admin Yang Mengelola Akun Customer		
Alien	Mengelola 2 Akun Customer	Rafa dan Salim
Amel Lia	Mengelola 3 Akun Customer	Atras dan Nabil dan Rafa
Christine	Mengelola 2 Akun Customer	Risa dan Nisa
Ela Setio	Mengelola 2 Akun Customer	Farida dan Nisa
Groot	Mengelola 2 Akun Customer	Wayan dan Agus
Jojo Nathan	Mengelola 3 Akun Customer	Nabil dan Wayan dan Dwi
Klaudi Banu	Mengelola 2 Akun Customer	Nabil dan Salim
Michael Alex	Mengelola 2 Akun Customer	Farida dan Risa
Niko Adri	Mengelola 3 Akun Customer	Atras dan Nabil dan Rafa
Wendy Anto	Mengelola 3 Akun Customer	Atras dan Wayan dan Dwi

d. Query keempat

- Fungsi: menampilkan ongkir perjalanan dan mengetahui harga per km nya (gofood)
- ```

SELECT CONCAT(RPAD("ID Order", 15," "), RPAD("Nama",11," "),
 RPAD("Ongkir Perjalanan", 23," "), "Jarak(km)") "Harga per km"
UNION
SELECT CONCAT(RPAD(me.ID_Order, 10," "),
 RPAD(CONCAT(c>NamaDepan, " ", IFNULL(c>NamaBelakang,"")),22," "),
 RPAD(me.TotalHarga-m.Harga, 21," "), mgr.Jarak) FROM Memesan me
JOIN Customer c USING (ID_Customer) JOIN Makanan m USING
(ID_Makanan) JOIN Mengantar mgr USING (ID_Order)
UNION
SELECT " "

```

UNION

SELECT CONCAT("Maka Harga per km adalah ",  
(me.TotalHarga-(m.Harga\*me.Qty))/mgr.Jarak) FROM Memesan me JOIN  
Makanan m USING (ID\_Makanan) JOIN Mengantar mgr USING  
(ID\_Order);

- Hasil eksekusi:

| Harga per km                  |                 |                   |           |
|-------------------------------|-----------------|-------------------|-----------|
| ID Order                      | Nama            | Ongkir Perjalanan | Jarak(km) |
| 00001                         | Atras Chandra   | 20000             | 10        |
| 00012                         | Atras Chandra   | 12000             | 6         |
| 00016                         | Atras Chandra   | 20000             | 10        |
| 00002                         | Nabil Rakha     | 10000             | 5         |
| 00013                         | Nabil Rakha     | 10000             | 5         |
| 00023                         | Rafa Hermansyah | 20000             | 10        |
| 00003                         | Rafa Hermansyah | 7000              | 3.5       |
| 00014                         | Rafa Hermansyah | 12000             | 6         |
| 00018                         | Rafa Hermansyah | 8000              | 4         |
| 00004                         | Wayan Saputra   | 8000              | 4         |
| 00005                         | Dwi Widya       | 10000             | 5         |
| 00015                         | Dwi Widya       | 12000             | 6         |
| 00017                         | Dwi Widya       | 8000              | 4         |
| 00006                         | Farida Setiawan | 20000             | 10        |
| 00019                         | Farida Setiawan | 22000             | 11        |
| 00007                         | Agus Setyo      | 11000             | 5.5       |
| 00022                         | Agus Setyo      | 10000             | 5         |
| 00020                         | Agus Setyo      | 14000             | 7         |
| 00008                         | Salim Putra     | 9000              | 4.5       |
| 00009                         | Risa Adinda     | 10000             | 5         |
| 00011                         | Nisa Ayu        | 10000             | 5         |
| 00010                         | Nisa Ayu        | 3000              | 1.5       |
| 00021                         | Nisa Ayu        | 7000              | 3.5       |
| Maka Harga per km adalah 2000 |                 |                   |           |

e. Query kelima

- Fungsi: Seluruh pemesanan dari gofood dan gomart

- SELECT "GoFood" AS "Seluruh Pemesanan"

UNION

SELECT CONCAT(RPAD(me.ID\_Order,8, " "),  
RPAD(CONCAT(c>NamaDepan, " ", IFNULL(c>NamaBelakang,"")), 20, "  
"), RPAD(mgr.TanggalAntar, 15, " "), me.TotalHarga) FROM Memesan  
me JOIN Customer c USING (ID\_Customer) JOIN Mengantar mgr USING  
(ID\_Order)

UNION

SELECT " "

```

UNION
SELECT "GoMart"
UNION
SELECT CONCAT(RPAD(mo.ID_mOrder,8, " "),
RPAD(CONCAT(c>NamaDepan, " ", IFNULL(c>NamaBelakang,"")), 20, "
"), RPAD(ma.TanggalAntar, 15, " "), mo.TotalHarga) FROM MartOrder mo
JOIN Customer c USING (ID_Customer) JOIN MartAntar ma USING
(ID_mOrder);

```

- Hasil eksekusi:

| Seluruh Pemesanan |                 |            |        |  |
|-------------------|-----------------|------------|--------|--|
| GoFood            |                 |            |        |  |
| 00001             | Atras Chandra   | 2021-04-03 | 50000  |  |
| 00012             | Atras Chandra   | 2021-06-06 | 87000  |  |
| 00016             | Atras Chandra   | 2021-08-08 | 32000  |  |
| 00002             | Nabil Rakha     | 2021-05-01 | 90000  |  |
| 00013             | Nabil Rakha     | 2021-03-04 | 65000  |  |
| 00023             | Rafa Hermansyah | 2021-04-04 | 100000 |  |
| 00003             | Rafa Hermansyah | 2021-07-07 | 57000  |  |
| 00014             | Rafa Hermansyah | 2021-03-04 | 35000  |  |
| 00018             | Rafa Hermansyah | 2021-03-05 | 43000  |  |
| 00004             | Wayan Saputra   | 2021-11-30 | 33000  |  |
| 00005             | Dwi Widya       | 2021-09-09 | 50000  |  |
| 00015             | Dwi Widya       | 2021-03-05 | 50000  |  |
| 00017             | Dwi Widya       | 2021-06-11 | 25000  |  |
| 00006             | Farida Setiawan | 2021-11-11 | 35000  |  |
| 00019             | Farida Setiawan | 2021-03-03 | 55000  |  |
| 00007             | Agus Setyo      | 2021-08-23 | 31000  |  |
| 00022             | Agus Setyo      | 2021-05-05 | 40000  |  |
| 00020             | Agus Setyo      | 2021-12-05 | 38000  |  |
| 00008             | Salim Putra     | 2021-12-12 | 39000  |  |
| 00009             | Risa Adinda     | 2021-01-01 | 40000  |  |
| 00011             | Nisa Ayu        | 2021-09-23 | 35000  |  |
| 00010             | Nisa Ayu        | 2021-11-03 | 33000  |  |
| 00021             | Nisa Ayu        | 2021-05-05 | 30000  |  |
| GoMart            |                 |            |        |  |
| M0001             | Atras Chandra   | 2021-10-18 | 20000  |  |
| M0002             | Nabil Rakha     | 2021-05-10 | 51000  |  |
| M0003             | Rafa Hermansyah | 2021-04-22 | 20000  |  |
| M0004             | Wayan Saputra   | 2021-03-07 | 30000  |  |
| M0005             | Dwi Widya       | 2021-08-09 | 18000  |  |
| M0006             | Farida Setiawan | 2021-09-11 | 128000 |  |
| M0007             | Agus Setyo      | 2021-11-11 | 44000  |  |
| M0008             | Salim Putra     | 2021-03-03 | 32000  |  |
| M0009             | Risa Adinda     | 2021-11-01 | 50000  |  |
| M0010             | Nisa Ayu        | 2021-09-09 | 81000  |  |

## 2. Transaction Control Language (TCL)

```
SET @@AUTOCOMMIT = 0;
```

```
-- saldo customer_1 sebelum menambah saldo
```

| ID_customer | SaldoGopay |
|-------------|------------|
| C0002       | 280000     |

```
-- terdapat seorang customer_1 yang menambah isi saldo Gopay sebanyak 100000
```

```
UPDATE Customer
```

```
SET SaldoGopay = saldogopay + 100000
```

```
WHERE ID_Customer = 'C0002';
```

```
-- Saldo Gopay setelah ditambah
```

| ID_customer | SaldoGopay |
|-------------|------------|
| C0002       | 380000     |

```
-- setelah mengisi/menambah saldo, database akan membuat save-point baru
```

```
SAVEPOINT SaldoTopUp;
```

```
-- kemudian, customer_1 mentransfer saldo kepada temannya sebanyak 100000
```

```
Update Customer
```

```
SET SaldoGopay = SaldoGopay - 100000
```

```
WHERE ID_Customer = 'C0002';
```

```
-- Saldo setelah melakukan transfer
```

| ID_customer | SaldoGopay |
|-------------|------------|
| C0002       | 280000     |

```
-- setelah melakukan transfer, ternyata saldo milik temannya tidak bertambah, namun
```

```
-- saldo customer 1 sudah berkurang. Setelah itu customer_1 menghubungi admin
```

```
-- Gojek dan menjelaskan permasalahannya. Mengetahui hal tersebut, admin
```

```
-- melakukan rollback ke save-point terakhir, yaitu setelah customer_1 mengisi saldo
```

```
-- terakhir kali
```

```
ROLLBACK TO SAVEPOINT SaldoTopUp;
```



-- dengan dilakukan rollback, maka customer\_1 mendapatkan saldo seperti seharusnya,

| ID_customer | SaldoGopay |
|-------------|------------|
| C0002       | 380000     |

-- dan ia melakukan transfer ulang untuk temannya, saldo customer\_1 berkurang

Update Customer

SET SaldoGopay = SaldoGopay - 100000

WHERE ID\_Customer = 'C0002';

-- bukti saldo berkurang

| ID_customer | SaldoGopay |
|-------------|------------|
| C0002       | 280000     |

-- transfer dari customer\_1 kepada temannya berhasil, maka saldo temannya bertambah

-- saldo sebelum ditransfer

| ID_customer | SaldoGopay |
|-------------|------------|
| C0003       | 340000     |

Update Customer

SET SaldoGopay = SaldoGopay + 100000

WHERE ID\_Customer = 'C0003';

-- saldo setelah ditransfer

| ID_customer | SaldoGopay |
|-------------|------------|
| C0003       | 440000     |

-- karena transfer sukses dan tidak ada permasalahan setelah 24 jam, maka database

--melakukan commit

COMMIT;

### 3. User Management, Privilege Management, & Views

#### a. Membuat 3 user

-- User ini berfungsi untuk manage toko  
CREATE USER 'ManagerToko'@'localhost'  
IDENTIFIED BY 'Nabil123';

-- User ini berfungsi untuk manage Shoppers  
CREATE USER 'ManagerShoppers'@'localhost'  
IDENTIFIED BY 'Rakha456';

-- User ini berfungsi untuk manage driver  
CREATE USER 'ManagerDriver'@'localhost'  
IDENTIFIED BY 'Dwitya789';

```
MariaDB [Gojek]> CREATE USER 'ManagerToko'@'localhost'
-> IDENTIFIED BY 'Nabil123';
Query OK, 0 rows affected (0.004 sec)
```

```
MariaDB [Gojek]> CREATE USER 'ManagerShoppers'@'localhost'
-> IDENTIFIED BY 'Rakha456';
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [Gojek]> CREATE USER 'ManagerDriver'@'localhost'
-> IDENTIFIED BY 'Dwitya789';
Query OK, 0 rows affected (0.002 sec)
```

#### b. Membuat 3 view

- CREATE OR REPLACE VIEW view\_penjualan\_toko (idOrder, namacus, tanggal, total)  
AS SELECT t.Nama Toko, GROUP\_CONCAT(CONCAT(c.NamaDepan, " ", IFNULL(c.NamaBelakang, ""))) Customer,  
GROUP\_CONCAT(ma.Nama) Makanan, COUNT(\*) Penjualan FROM memesan m JOIN Toko t USING (ID\_Toko) JOIN Customer c USING (ID\_Customer) JOIN Makanan ma USING (ID\_Makanan) GROUP BY 1;

```
MariaDB [Gojek]> CREATE OR REPLACE VIEW view_penjualan_toko (idOrder, namacus, tanggal, total)
-> AS SELECT t.Nama Toko, GROUP_CONCAT(CONCAT(c.NamaDepan, " ", IFNULL(c.NamaBelakang, ""))) Customer, GROUP_CONCAT(ma.Nama) Makanan, COUNT(*) Penjualan FROM Memesan m JOIN Toko t USING (ID_Toko) JOIN Customer c USING (ID_Customer) JOIN Makanan ma USING (ID_Makanan) GROUP BY 1;
Query OK, 0 rows affected (0.004 sec)
```

View ini berfungsi untuk melihat kinerja penjualan toko.

- CREATE OR REPLACE VIEW view\_kinerja\_shopper (idshopper, nama, namacus, rateshopper, namabarang, namamart)  
AS SELECT sh.ID\_Shopper ID, CONCAT(sh.NamaDepan, " ", IFNULL(sh.NamaBelakang, "")) Nama, CONCAT(c.NamaDepan, " ",

```
IFNULL(c>NamaBelakang,""))>NamaCus, ma.RateShopper Rating,
b>Nama, s>NamaMart FROM MartAntar ma JOIN MartOrder mo USING
(ID_mOrder) JOIN Shopper sh USING (ID_Shopper) JOIN Customer c on
(c.ID_Customer = ma.customerid) JOIN Barang b ON (b.ID_Barang =
ma.ID_Barang) JOIN Supermart s ON (s.ID_sMart = mo.ID_Mart);
```

```
MariaDB [Gojek]> CREATE OR REPLACE VIEW view_kinerja_shopper (idshopper, nama, namacus, ra
teshopper, namabarang, namamart)
-> AS SELECT sh.ID_Shopper ID, CONCAT(sh>NamaDepan, " ", IFNULL(sh>NamaBelakang,"")) N
ama, CONCAT(c>NamaDepan, " ", IFNULL(c>NamaBelakang,""))>NamaCus, ma.RateShopper Rating, b
>Nama, s>NamaMart FROM MartAntar ma JOIN MartOrder mo USING (ID_mOrder) JOIN Shopper sh US
ING (ID_Shopper) JOIN Customer c on (c.ID_Customer = ma.customerid) JOIN Barang b ON (b.ID
_Barang = ma.ID_Barang) JOIN Supermart s ON (s.ID_sMart = mo.ID_Mart);
Query OK, 0 rows affected (0.002 sec)
```

View ini berfungsi untuk melihat kinerja shopper.

- CREATE OR REPLACE VIEW view\_kinerja\_Driver (iddriver, namadriver, namacus, namamakanan, rating)  
AS SELECT d.ID\_Driver, CONCAT(d>NamaDepan, " ",  
IFNULL(d>NamaBelakang,"")), CONCAT(c>NamaDepan, " ",  
IFNULL(c>NamaBelakang,"")), m>Nama, ma.RateDriver FROM Mengantar  
ma JOIN Driver d USING (ID\_Driver) JOIN Customer c USING  
(ID\_Customer) JOIN Makanan m USING (ID\_Makanan);

```
MariaDB [Gojek]> CREATE OR REPLACE VIEW view_kinerja_Driver (iddriver, namadriver, namacu
s, namamakanan, rating)
-> AS SELECT d.ID_Driver, CONCAT(d>NamaDepan, " ", IFNULL(d>NamaBelakang,"")), CONCAT
(c>NamaDepan, " ", IFNULL(c>NamaBelakang,"")), m>Nama, ma.RateDriver FROM Mengantar ma JO
IN Driver d USING (ID_Driver) JOIN Customer c USING (ID_Customer) JOIN Makanan m USING (I
D_Makanan);
Query OK, 0 rows affected (0.005 sec)
```

View ini berfungsi untuk melihat kinerja driver.

c. Memberikan privilege kepada user.

-- memantau kinerja penjualan toko

GRANT SELECT

ON view\_penjualan\_toko

TO 'ManagerToko'@'localhost';

-- memantau kinerja shoppers

GRANT SELECT

ON view\_kinerja\_shopper

TO 'ManagerShoppers'@'localhost';

-- memantau kinerja driver

GRANT SELECT

ON view\_kinerja\_driver

TO 'ManagerDriver'@'localhost';

```

MariaDB [Gojek]> -- memantau kinerja penjualan toko
MariaDB [Gojek]> GRANT SELECT
 -> ON view_penjualan_toko
 -> TO 'ManagerToko'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [Gojek]>
MariaDB [Gojek]> -- memantau kinerja shoppers
MariaDB [Gojek]> GRANT SELECT
 -> ON view_kinerja_shopper
 -> TO 'ManagerShoppers'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [Gojek]>
MariaDB [Gojek]> -- memantau kinerja driver
MariaDB [Gojek]> GRANT SELECT
 -> ON view_kinerja_driver
 -> TO 'ManagerDriver'@'localhost';
Query OK, 0 rows affected (0.001 sec)

```

Bukti user ManagerToko melihat kinerja toko.

```
mysql -u ManagerToko -p
```

```
Enter password: ****
```

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
```

```
Your MariaDB connection id is 17
```

```
MariaDB [gojek]> SELECT idOrder "ID Order", namacus "Nama Customer", tanggal "Tanggal", total "Total" FROM view_penjualan_toko;
```

| ID Order       | Nama Customer                                      | Tanggal                                             | Total |
|----------------|----------------------------------------------------|-----------------------------------------------------|-------|
| Angkringan     | Agus Setyo,Dwi Widya                               | Nasi Kucing,Nasi Putih                              | 2     |
| Bakery Cake    | Rafa Hermansyah,Nabil Rakha                        | Kue Hantu,Kue Kuean                                 | 2     |
| KFC            | Wayan Saputra,Rafa Hermansyah,Nisa Ayu,Nisa Ayu    | EsKrim Cone,EsKrim Coklat,EsKrim Coklat,EsKrim Cone | 4     |
| MCD            | Atras Chandra                                      | Ayam Goyeng                                         | 1     |
| Nasi Goreng 99 | Risa Adinda,Agus Setyo,Farida Setiawan             | Nasi Uduk,Nasi Uduk,Nasi Goreng Special             | 3     |
| PizzaHut       | Rafa Hermansyah,Nisa Ayu,Nabil Rakha,Atras Chandra | Pizza Aja,Kopi Hitam,Pizza Aja,Pizza Jamur          | 4     |
| RotiBakar88    | Agus Setyo                                         | Roti Bakar Coklat                                   | 1     |
| Sate Senayan   | Rafa Hermansyah,Salim Putra                        | Es Teh Panas,Sate Ayam                              | 2     |
| Warteg         | Atras Chandra,Farida Setiawan                      | Tempe Orek,Telur Pedas                              | 2     |
| Wendys         | Dwi Widya,Dwi Widya                                | Milkshake,Kerupuk Kulit                             | 2     |

Bukti user ManagerShoppers melihat kinerja shopper.

```
mysql -u ManagerShoppers -p
```

```
Enter password: ****
```

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
```

```
Your MariaDB connection id is 18
```

```
Server version: 10.4.20-MariaDB mariadb.org binary distribution
```

| ID Shopper | Nama Shopper  | Customer yg Dilayani | Kinerja | Barang                  | Asal Supermart  |
|------------|---------------|----------------------|---------|-------------------------|-----------------|
| SH001      | Awil Sanves   | Atras Chandra        | 3       | BEVERAGES Frestea       | Emerald Bintaro |
| SH003      | Anton Nius    | Nabil Rakha          | 4       | BEVERAGES Marjan Squash | Lotte Mart      |
| SH005      | Budi Saputra  | Rafa Hermansyah      | 3       | BEVERAGES Nutriboost    | Hypermart       |
| SH007      | Alex Xander   | Wayan Saputra        | 5       | BEVERAGES Mogu Mogu     | Sayurbox        |
| SH009      | Bintang       | Dwi Widya            | 4       | BEVERAGES Nata De Coco  | Market City     |
| SH011      | Ko Achil      | Farida Setiawan      | 2       | BEAUTY Wardah           | Alfamart        |
| SH013      | Liban         | Agus Setyo           | 3       | BEVERAGES Sidomuncul    | Alfamidi        |
| SH015      | Sarta Shalhan | Salim Putra          | 5       | SNACKS Chitato          | Circle K        |
| SH017      | SEN Tenz      | Risa Adinda          | 4       | SNACKS Betadine         | Guardian        |
| SH019      | Caroline      | Nisa Ayu             | 5       | SNACKS Frozz            | Watsons         |

Bukti user ManagerDriver melihat kinerja driver.

```
mysql -u ManagerDriver -p
Enter password: *****
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 19
Server version: 10.4.20-MariaDB mariadb.org binary distribution
```

| ID Driver | Nama Driver   | Customer yg Dilayani | Makanan             | Kinerja |
|-----------|---------------|----------------------|---------------------|---------|
| D0001     | Eko Putra     | Atras Chandra        | Ayam Goyeng         | 3       |
| D0001     | Eko Putra     | Atras Chandra        | Tempe Orek          | 3       |
| D0002     | Indah Bulan   | Atras Chandra        | Pizza Jamur         | 5       |
| D0002     | Indah Bulan   | Nabil Rakha          | Pizza Aja           | 4       |
| D0005     | Dian Cahaya   | Nabil Rakha          | Kue Kuean           | 4       |
| D0001     | Eko Putra     | Rafa Hermansyah      | Es Teh Panas        | 5       |
| D0003     | Ade Irwan     | Rafa Hermansyah      | Kue Hantu           | 5       |
| D0005     | Dian Cahaya   | Rafa Hermansyah      | Pizza Aja           | 2       |
| D0007     | Ratna Devi    | Rafa Hermansyah      | EsKrim Coklat       | 4       |
| D0004     | Susilo Irwan  | Wayan Saputra        | EsKrim Cone         | 4       |
| D0004     | Susilo Irwan  | Dwi Widya            | Nasi Putih          | 2       |
| D0005     | Dian Cahaya   | Dwi Widya            | Kerupuk Kulit       | 5       |
| D0008     | Hermanto Syah | Dwi Widya            | Milkshake           | 3       |
| D0005     | Dian Cahaya   | Farida Setiawan      | Nasi Goreng Special | 5       |
| D0006     | Slamet Darma  | Farida Setiawan      | Telor Pedas         | 2       |
| D0007     | Ratna Devi    | Agus Setyo           | Nasi Kucing         | 2       |
| D0009     | Sinta Sari    | Agus Setyo           | Nasi Uduk           | 3       |
| D0009     | Sinta Sari    | Agus Setyo           | Roti Bakar Coklat   | 2       |
| D0008     | Hermanto Syah | Salim Putra          | Sate Ayam           | 3       |
| D0009     | Sinta Sari    | Risa Adinda          | Nasi Uduk           | 3       |
| D0001     | Eko Putra     | Nisa Ayu             | EsKrim Cone         | 5       |
| D0008     | Hermanto Syah | Nisa Ayu             | EsKrim Coklat       | 5       |
| D0010     | Rahmat Basuki | Nisa Ayu             | Kopi Hitam          | 4       |

#### 4. Stored Routines & Triggers

##### a. Stored procedure

- Mencari nama barang sesuai kategori

```
DELIMITER //
CREATE OR REPLACE PROCEDURE search_barang(IN kategori
VARCHAR(255))
BEGIN
 DECLARE namaBarang VARCHAR (100);
 DECLARE mart VARCHAR (100);
 DECLARE totalBarang, ctr, stokbrg INTEGER;
 DECLARE c_Barang CURSOR FOR SELECT nama, stok, s>NamaMart
FROM Barang JOIN Supermart s USING (ID_sMart) WHERE nama LIKE
CONCAT(kategori, '%');
 OPEN c_Barang;
 CREATE TEMPORARY TABLE List(data VARCHAR(255));
 SELECT COUNT(nama) INTO totalBarang FROM Barang WHERE nama
LIKE CONCAT(kategori, '%');
 IF(totalBarang = 0) THEN
 INSERT INTO List(data) VALUES ('Barang Tidak ada di Supermart
manapun');
 ELSE
 INSERT INTO List(data) VALUES (CONCAT('Ditemukan ', totalBarang, '
di Supermart beserta stoknya berikut'));
 INSERT INTO List(data) VALUES(' ');
 INSERT INTO List(data) VALUES (CONCAT(RPAD(' Nama
Barang',18," "), RPAD('stok',8," "), 'Nama Supermart'));
 SET ctr = 0;
 WHILE ctr < totalBarang DO
 FETCH c_Barang INTO namaBarang, stokbrg, mart;
 INSERT INTO List(data) VALUES
(CONCAT(RPAD(namaBarang,19," "), RPAD(stokbrg,8," "), mart));
 SET ctr = ctr + 1;
 END WHILE;
 END IF;
 CLOSE c_Barang;
 SELECT data "Search Barang "FROM List;
 DROP TEMPORARY TABLE List;
END //
DELIMITER ;
```

```

MariaDB [gojek]> DELIMITER //
MariaDB [gojek]> CREATE OR REPLACE PROCEDURE search_barang(IN kategori VARCHAR(255))
-> BEGIN
-> DECLARE namaBarang VARCHAR (100);
-> DECLARE mart VARCHAR (100);
-> DECLARE totalBarang, ctr, stokbrg INTEGER;
-> DECLARE c_Barang CURSOR FOR SELECT nama, stok, s>NamaMart FROM Barang JOIN Supermart s USING (ID_sMart) WHERE nama LIKE CONCAT(kategori, '%');
-> OPEN c_Barang;
-> CREATE TEMPORARY TABLE List(data VARCHAR(255));
-> SELECT COUNT(nama) INTO totalBarang FROM Barang WHERE nama LIKE CONCAT(kategori, '%');
-> IF(totalBarang = 0) THEN
-> INSERT INTO List(data) VALUES ('Barang Tidak ada di Supermart manapun');
-> ELSE
-> INSERT INTO List(data) VALUES (CONCAT('Ditemukan ', totalBarang, ' di Supermart beserta stoknya berikut'));
-> INSERT INTO List(data) VALUES(' ');
-> INSERT INTO List(data) VALUES (CONCAT(RPAD(' Nama Barang',18," "), RPAD('stok',8," "), 'Nama Supermart'));
-> SET ctr = 0;
-> WHILE ctr < totalBarang DO
-> FETCH c_Barang INTO namaBarang, stokbrg, mart;
-> INSERT INTO List(data) VALUES (CONCAT(RPAD(namaBarang,19," "), RPAD(stokbrg,8," "), mart));
-> SET ctr = ctr + 1;
-> END WHILE;
-> END IF;
-> CLOSE c_Barang;
-> SELECT data "Search Barang "FROM List;
-> DROP TEMPORARY TABLE List;
-> END //
Query OK, 0 rows affected (0.030 sec)
MariaDB [gojek]> DELIMITER ;

```

```

MariaDB [gojek]> CALL search_barang('SNACKS');
+-----+
| Search Barang |
+-----+
| Ditemukan 3 di Supermart beserta stoknya berikut |
|
| Nama Barang stok Nama Supermart
| SNACKS Chitato 35 Circle K
| SNACKS Betadine 15 Guardian
| SNACKS Frozz 20 Watsons
+-----+

MariaDB [gojek]> CALL search_barang('Mangan');
+-----+
| Search Barang |
+-----+
| Barang Tidak ada di Supermart manapun |
+-----+
1 row in set (0.002 sec)

```

- Mencari seluruh pegawai

```
DELIMITER //
```

```
CREATE OR REPLACE PROCEDURE search_pegawai(IN id CHAR(5))
```

```
BEGIN
```

```
 DECLARE fullname VARCHAR(255);
```

```
 DECLARE valid INTEGER;
```

```
 SELECT CASE
```

```
 WHEN SUBSTRING(id, 1,1) = "D" AND LENGTH(id) = 5 THEN
```

```
(SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, "
```

```

)),15, " "), RPAD (email,16, " "),
NoTelpon) FROM Driver WHERE ID_Driver = id)
 WHEN SUBSTRING(id, 1,1) = "A" AND LENGTH(id) = 5 THEN
(SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, "
)),15, " "), RPAD (email,20, " "),
NoTelpon) FROM Admin WHERE ID_Admin = id)
 WHEN SUBSTRING(id, 1,1) = "S" AND LENGTH(id) = 5 THEN
(SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, "
)),15, " "), RPAD (s>NamaMart,20,
" "), NoTelpon) FROM Shopper JOIN Supermart s USING (ID_sMart) WHERE
ID_Shopper = id)
 ELSE 'Tidak Ada Pegawai Dengan ID tersebut'
 END INTO fullname;
CREATE TEMPORARY TABLE pegawai(data VARCHAR(255));
INSERT INTO pegawai(data) VALUES(fullname);
SELECT data AS "Informasi Pegawai" FROM pegawai;
DROP TEMPORARY TABLE pegawai;
END //
DELIMITER ;

```

```

MariaDB [gojek]> CREATE OR REPLACE PROCEDURE search_pegawai(IN id CHAR(5))
-> BEGIN
-> DECLARE fullname VARCHAR(255);
-> DECLARE valid INTEGER;
-> SELECT CASE
-> WHEN SUBSTRING(id, 1,1) = "D" AND LENGTH(id) = 5 THEN (SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, " "),15, " "), RPAD
(email,16, " "), NoTelpon) FROM Driver WHERE ID_Driver = id)
-> WHEN SUBSTRING(id, 1,1) = "A" AND LENGTH(id) = 5 THEN (SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, " "),15, " "), RPAD
(email,20, " "), NoTelpon) FROM Admin WHERE ID_Admin = id)
-> WHEN SUBSTRING(id, 1,1) = "S" AND LENGTH(id) = 5 THEN (SELECT CONCAT(RPAD(CONCAT(Namadepan, " ", IFNULL>NamaBelakang, " "),15, " "), RPAD
(s>NamaMart,20, " "), NoTelpon) FROM Shopper JOIN Supermart s USING (ID_sMart) WHERE
ID_Shopper = id)
-> ELSE 'Tidak Ada Pegawai Dengan ID tersebut'
-> END INTO fullname;
-> CREATE TEMPORARY TABLE pegawai(data VARCHAR(255));
-> INSERT INTO pegawai(data) VALUES(fullname);
-> SELECT data AS "Informasi Pegawai" FROM pegawai;
-> DROP TEMPORARY TABLE pegawai;
-> END //
Query OK, 0 rows affected (0.013 sec)
MariaDB [gojek]> DELIMITER ;

```

```

MariaDB [gojek]> CALL search_pegawai('D0001');
+-----+
| Informasi Pegawai |
+-----+
| Eko Putra Eko@gmail.com 081239182839 |
+-----+

```

```

MariaDB [gojek]> CALL search_pegawai('A0001');
+-----+
| Informasi Pegawai |
+-----+
| Wendy Anto Wendy@gmail.com 081325823215 |
+-----+

```



```

MariaDB [gojek]> CALL search_pegawai('SH001');
+-----+
| Informasi Pegawai |
+-----+
| Awi Sanves Emerald Bintaro 081215286032 |
+-----+

MariaDB [gojek]> CALL search_pegawai('e001');
+-----+
| Informasi Pegawai |
+-----+
| Tidak Ada Pegawai Dengan ID tersebut |
+-----+

```

b. Stored function

- Membuat tingkatan pada customer berdasarkan total harga di GoFood

DELIMITER //

CREATE FUNCTION CustomerLevel(TotalHarga INTEGER) RETURNS  
VARCHAR(255)

BEGIN

DECLARE CustomerLevel VARCHAR(255);

IF TotalHarga < 50000 THEN

SET CustomerLevel = 'Bronze';

ELSEIF TotalHarga < 100000 THEN

SET CustomerLevel = 'Silver';

ELSEIF TotalHarga < 150000 THEN

SET CustomerLevel = 'Gold';

ELSEIF TotalHarga < 200000 THEN

SET CustomerLevel = 'Platinum';

ELSEIF TotalHarga >= 200000 THEN

SET CustomerLevel = 'Radiant';

END IF;

RETURN CustomerLevel;

END //

DELIMITER ;

```

MariaDB [gojek]> DELIMITER //
MariaDB [gojek]> CREATE FUNCTION CustomerLevel(TotalHarga INTEGER) RETURNS VARCHAR(255)
-> BEGIN
-> DECLARE CustomerLevel VARCHAR(255);
-> IF TotalHarga < 50000 THEN
-> SET CustomerLevel = 'Bronze';
-> ELSEIF TotalHarga < 100000 THEN
-> SET CustomerLevel = 'Silver';
-> ELSEIF TotalHarga < 150000 THEN
-> SET CustomerLevel = 'Gold';
-> ELSEIF TotalHarga < 200000 THEN
-> SET CustomerLevel = 'Platinum';
-> ELSEIF TotalHarga >= 200000 THEN
-> SET CustomerLevel = 'Radiant';
-> END IF;
-> RETURN CustomerLevel;
-> END //
Query OK, 0 rows affected (0.005 sec)

```

```

SELECT CONCAT(c>NamaDepan, " ", IFNULL(c>NamaBelakang,"")) "Nama
Customer", CustomerLevel(SUM(TotalHarga)) "Customer Level" FROM
MEMESAN JOIN CUSTOMER C USING (ID_Customer) GROUP BY 1 ORDER
BY SUM(TOTALHARGA) DESC;

```

| Nama Customer   | Customer Level |
|-----------------|----------------|
| Rafa Hermansyah | Radiant        |
| Atras Chandra   | Platinum       |
| Nabil Rakha     | Platinum       |
| Dwi Widya       | Gold           |
| Agus Setyo      | Gold           |
| Nisa Ayu        | Silver         |
| Farida Setiawan | Silver         |
| Risa Adinda     | Bronze         |
| Salim Putra     | Bronze         |
| Wayan Saputra   | Bronze         |

- Fungsi untuk mengkategorikan shopper berdasarkan rating yg diberikan customer, mana yg layak dipecat atau tidak.

```

DELIMITER //
CREATE FUNCTION ShopperRate(Rate INTEGER) RETURNS VARCHAR(255)
BEGIN
 DECLARE Klasifikasi VARCHAR(255);
 IF Rate = 0 THEN
 SET Klasifikasi = 'Sangat Tidak Aman';

```

```

ELSEIF Rate = 1 THEN
 SET Klasifikasi = 'Tidak Aman';
ELSEIF Rate = 2 THEN
 SET Klasifikasi = 'Sedikit Tidak Aman';
ELSEIF Rate = 3 THEN
 SET Klasifikasi = 'Aman';
ELSEIF Rate = 4 THEN
 SET Klasifikasi = 'Lumayan Aman';
ELSEIF Rate = 5 THEN
 SET Klasifikasi = 'SIUUUUUUUUUUUUUUUUUUUU';
END IF;
RETURN Klasifikasi;
END //
DELIMITER ;

```

```

MariaDB [gojek]> DELIMITER //
MariaDB [gojek]> CREATE FUNCTION ShopperRate(Rate INTEGER) RETURNS VARCHAR(255)
-> BEGIN
-> DECLARE Klasifikasi VARCHAR(255);
-> IF Rate = 0 THEN
-> SET Klasifikasi = 'Sangat Tidak Aman';
-> ELSEIF Rate = 1 THEN
-> SET Klasifikasi = 'Tidak Aman';
-> ELSEIF Rate = 2 THEN
-> SET Klasifikasi = 'Sedikit Tidak Aman';
-> ELSEIF Rate = 3 THEN
-> SET Klasifikasi = 'Aman';
-> ELSEIF Rate = 4 THEN
-> SET Klasifikasi = 'Lumayan Aman';
-> ELSEIF Rate = 5 THEN
-> SET Klasifikasi = 'SIUUUUUUUUUUUUUUUUUUUU';
-> END IF;
-> RETURN Klasifikasi;
-> END //
Query OK, 0 rows affected (0.005 sec)

```

```

SELECT mo.ID_Shopper "ID Shopper", CONCAT(sh>NamaDepan, " ",
IFNULL(sh>NamaBelakang,"")) "Nama Shopper",
ShopperRate(SUM(RateShopper)/COUNT(*)) "Rata-Rata Kinerja" FROM
MartAntar join martorder mo using (id_morder) JOIN Shopper sh
ON(sh.ID_Shopper = mo.ID_Shopper) GROUP BY 1 ORDER BY

```

SUM(RateShopper)/COUNT(\*) DESC;

| ID Shopper | Nama Shopper  | Rata-Rata Kinerja    |
|------------|---------------|----------------------|
| SH019      | Caroline      | SIUUUUUUUUUUUUUUUUUU |
| SH007      | Alex Xander   | SIUUUUUUUUUUUUUUUUUU |
| SH015      | Sarta Shalhan | SIUUUUUUUUUUUUUUUUUU |
| SH009      | Bintang       | Lumayan Aman         |
| SH017      | SEN Tenz      | Lumayan Aman         |
| SH003      | Anton Nius    | Lumayan Aman         |
| SH001      | Awi Sanves    | Aman                 |
| SH005      | Budi Saputra  | Aman                 |
| SH013      | Liban         | Aman                 |
| SH011      | Ko Achil      | Sedikit Tidak Aman   |

c. Trigger

- Untuk membuat riwayat terkait top up gopay yg dilakukan customer

DELIMITER //

CREATE TABLE TopUpGopay(

ID\_Customer CHAR(5),

NamaDepan VARCHAR(30),

NamaBelakang VARCHAR(30),

SaldoGopay INTEGER,

SaldoTopUp INTEGER,

TopUpDate Date

);

CREATE OR REPLACE TRIGGER GopayTopUp

AFTER UPDATE ON Customer FOR EACH ROW

BEGIN

INSERT INTO TopUpGopay

VALUES (NEW.ID\_Customer, NEW>NamaDepan,

NEW>NamaBelakang, OLD.SaldoGopay,

(NEW.SaldoGopay-OLD.SaldoGopay), current\_date());

END //

DELIMITER ;

```

MariaDB [gojek]> DELIMITER //
MariaDB [gojek]> CREATE TABLE TopUpGopay(
-> ID_Customer CHAR(5),
-> NamaDepan VARCHAR(30),
-> NamaBelakang VARCHAR(30),
-> SaldoGopay INTEGER,
-> SaldoTopUp INTEGER,
-> TopUpDate Date
->);
->
-> CREATE OR REPLACE TRIGGER GopayTopUp
-> AFTER UPDATE ON Customer FOR EACH ROW
-> BEGIN
-> INSERT INTO TopUpGopay
-> VALUES (NEW.ID_Customer, NEW>NamaDepan, NEW>NamaBelakang, OLD.SaldoGopay, (NEW.SaldoGopay-OLD.SaldoGopay), current_date());
-> END //
Query OK, 0 rows affected (0.017 sec)

```

```

UPDATE customer
SET saldogopay = 210000
WHERE id_customer = 'c0001';
SELECT * FROM TopUpGopay;

```

```

MariaDB [gojek]> update customer
-> set saldogopay = 210000
-> where id_customer = 'c0001';
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

```

```

MariaDB [gojek]> Select * from TopUpGopay;

```

| ID_Customer | NamaDepan | NamaBelakang | SaldoGopay | SaldoTopUp | TopUpDate  |
|-------------|-----------|--------------|------------|------------|------------|
| C0001       | Atras     | Chandra      | 200000     | 10000      | 2021-12-13 |

- Untuk membuat riwayat terkait perubahan gaji admin.

```

DELIMITER //
CREATE TABLE GajiUpdate(
ID_Admin CHAR(5),
NamaDepan VARCHAR(30),
NamaBelakang VARCHAR(30),
Gaji_awal INTEGER,
Gaji_akhir INTEGER,
PerubahanGaji VARCHAR(30),
Tanggal DATE
);

```

```

CREATE OR REPLACE TRIGGER UpdateGaji
AFTER UPDATE ON Admin FOR EACH ROW
BEGIN
IF NEW.Gaji > OLD.Gaji THEN
INSERT INTO GajiUpdate

```

```

VALUES(NEW.ID_Admin, NEW>NamaDepan, NEW>NamaBelakang,
OLD.Gaji, NEW.Gaji, CONCAT("+",
TRUNCATE((((NEW.Gaji-OLD.Gaji)/OLD.Gaji)*100),2), "%"),
current_date());
ELSE
INSERT INTO GajiUpdate
VALUES(NEW.ID_Admin, NEW>NamaDepan, NEW>NamaBelakang,
OLD.Gaji, NEW.Gaji, CONCAT("-",
TRUNCATE((((OLD.Gaji-NEW.Gaji)/OLD.Gaji)*100),2), "%"),
current_date());
END IF;
END //
DELIMITER ;

```

```

MariaDB [gojek]> DELIMITER //
MariaDB [gojek]> CREATE TABLE GajiUpdate(
-> ID_Admin CHAR(5),
-> NamaDepan VARCHAR(30),
->>NamaBelakang VARCHAR(30),
-> Gaji_awal INTEGER,
-> Gaji_akhir INTEGER,
-> PerubahanGaji VARCHAR(30),
-> Tanggal DATE
->);
->
-> CREATE OR REPLACE TRIGGER UpdateGaji
-> AFTER UPDATE ON Admin FOR EACH ROW
-> BEGIN
-> IF NEW.Gaji > OLD.Gaji THEN
-> INSERT INTO GajiUpdate
-> VALUES(NEW.ID_Admin, NEW>NamaDepan, NEW>NamaBelakang, OLD.Gaji, NEW.Gaji, CONCAT("+", TRUNCATE((((NEW.Gaji-OLD.Gaji)/OLD.Gaji)*100),2), "%"), current_date
());
-> ELSE
-> INSERT INTO GajiUpdate
-> VALUES(NEW.ID_Admin, NEW>NamaDepan, NEW>NamaBelakang, OLD.Gaji, NEW.Gaji, CONCAT("-", TRUNCATE((((OLD.Gaji-NEW.Gaji)/OLD.Gaji)*100),2), "%"), current_date
());
-> END IF;
-> END //
Query OK, 0 rows affected (0.025 sec)

```

```

UPDATE admin
SET gaji = 3500000
WHERE id_admin = 'A0001';
SELECT * FROM GajiUpdate;

```

```

MariaDB [gojek]> UPDATE admin
-> SET gaji = 3500000
-> WHERE id_admin = 'A0001';
Query OK, 1 row affected (0.003 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [gojek]> SELECT * FROM GajiUpdate;
+-----+-----+-----+-----+-----+-----+
| ID_Admin |>NamaDepan |>NamaBelakang | Gaji_awal | Gaji_akhir | PerubahanGaji | Tanggal |
+-----+-----+-----+-----+-----+-----+
| A0001 | Wendy | Anto | 3000000 | 3500000 | +16.66% | 2021-12-13 |
+-----+-----+-----+-----+-----+-----+

```

```

UPDATE admin
SET gaji = 3000000
WHERE id_admin = 'A0001';
SELECT * FROM GajiUpdate;

```

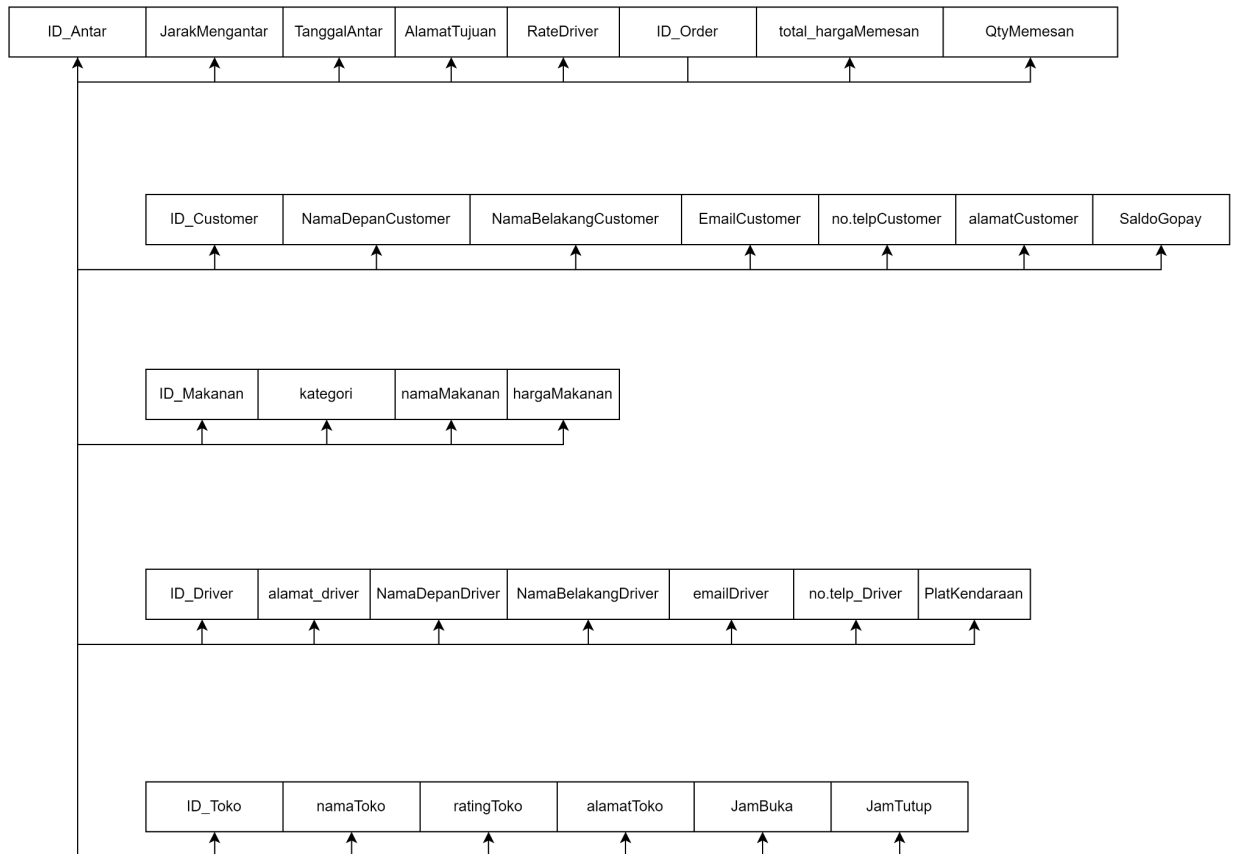
```
MariaDB [gojek]> UPDATE admin
 -> SET gaji = 3000000
 -> WHERE id_admin = 'A0001';
Query OK, 1 row affected (0.002 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [gojek]> SELECT * FROM GajiUpdate;
```

| ID_Admin | NamaDepan | NamaBelakang | Gaji_awal | Gaji_akhir | PerubahanGaji | Tanggal    |
|----------|-----------|--------------|-----------|------------|---------------|------------|
| A0001    | Wendy     | Anto         | 3000000   | 3500000    | +16.66%       | 2021-12-13 |
| A0001    | Wendy     | Anto         | 3500000   | 3000000    | -14.28%       | 2021-12-13 |

## 5. Normalisasi

- Normalisasi GoFood
  - Data Requirements

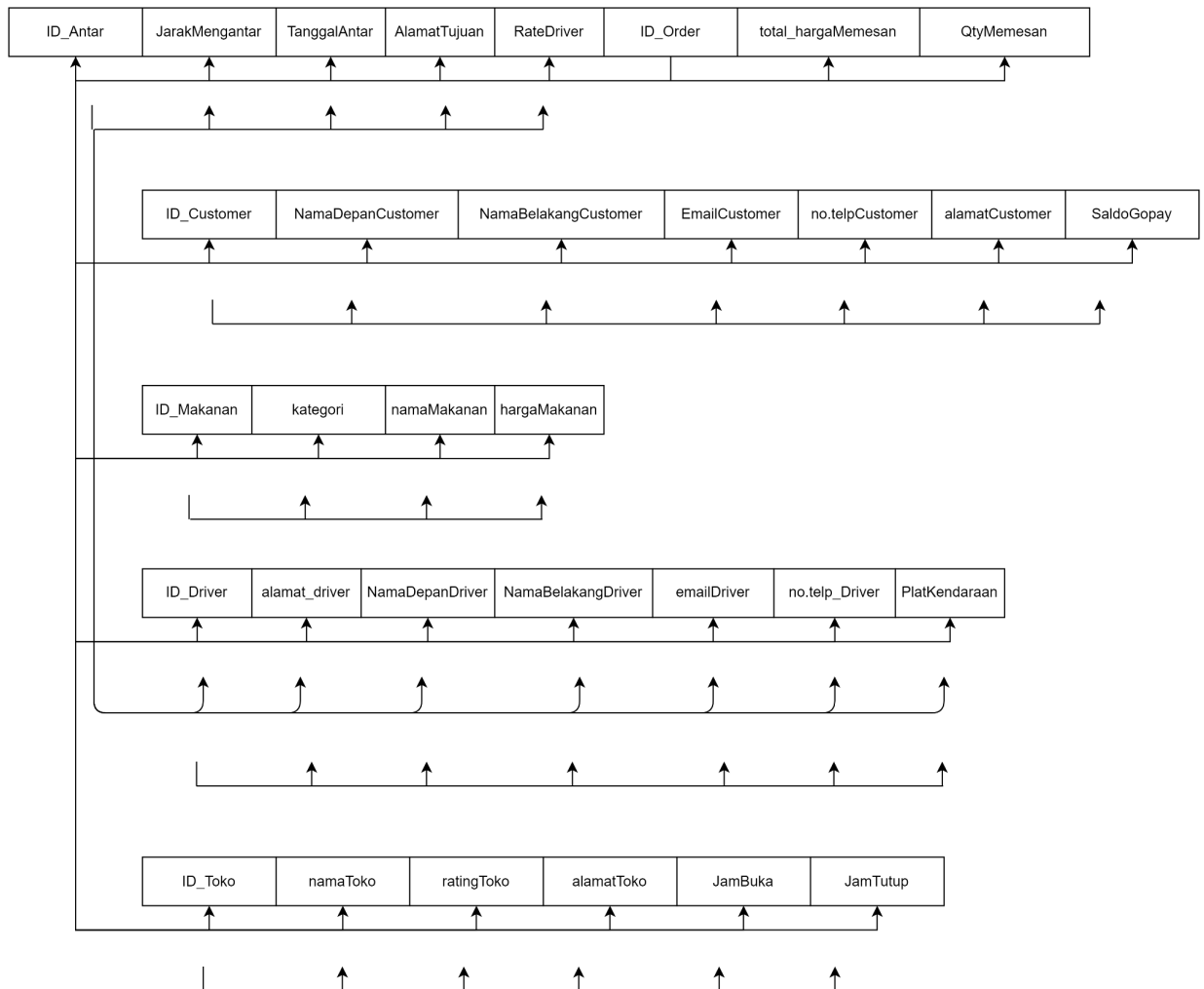


Dari data requirements tersebut yang menjadi Primary Key awal adalah ID\_Order

ID\_Order → ID\_Antar, JarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, total\_hargaMemesan, QtyMemesan, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay, ID\_Makanan, kategori, namaMakanan, hargaMakanan, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan, ID\_Toko, namaToko, ratingToko, alamatToko, JamBuka, JamTutup



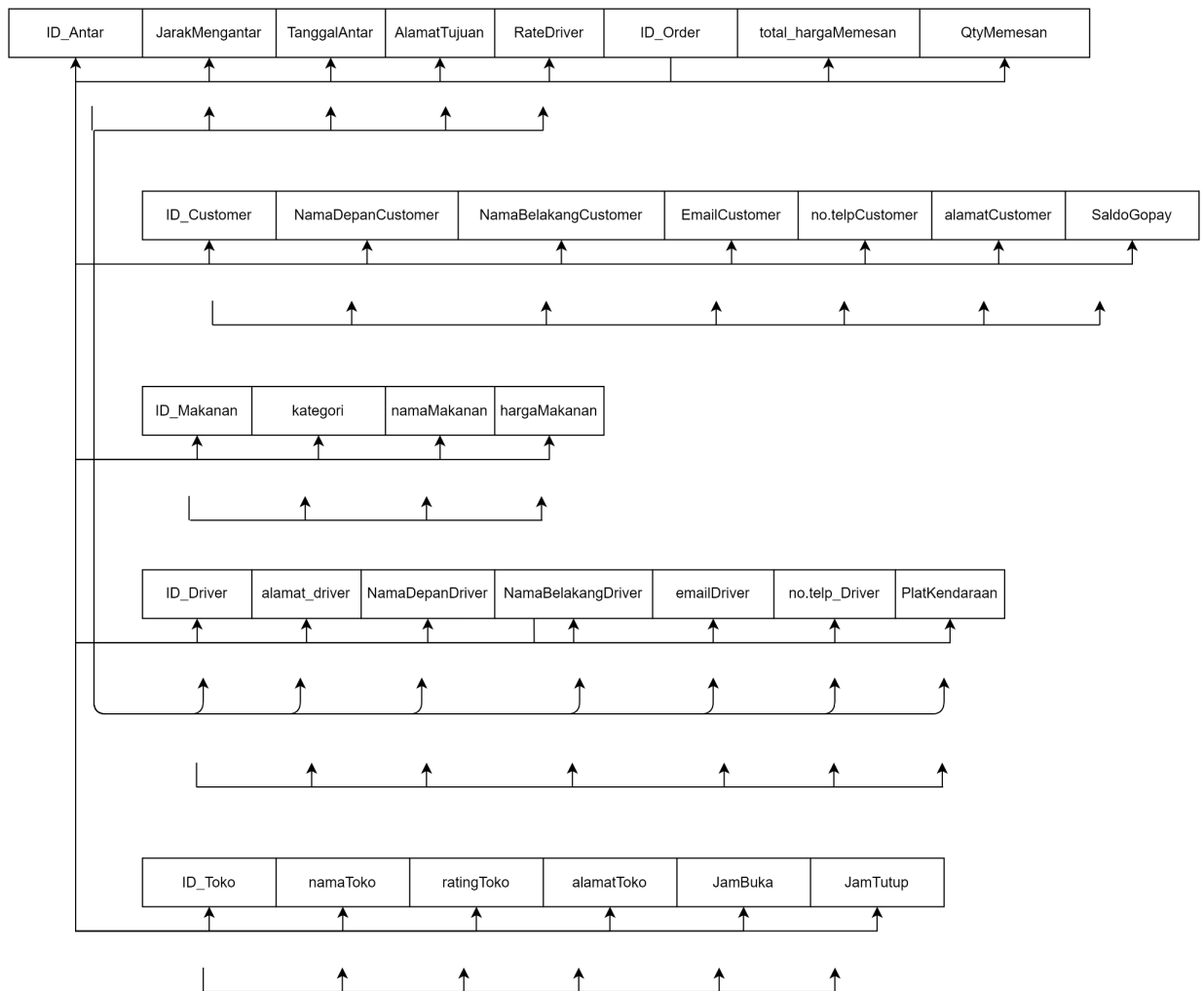
○ 1NF



Dari ID\_Order, tidak ada kolom yang bernilai multivalue maka dari itu tidak ada tambahan Primary Key.

ID\_Order → ID\_Antar, JarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, total\_hargaMemesan, QtyMemesan, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay, ID\_Makanan, kategori, namaMakanan, hargaMakanan, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan, ID\_Toko, namaToko, ratingToko, alamatToko, JamBuka, JamTutup

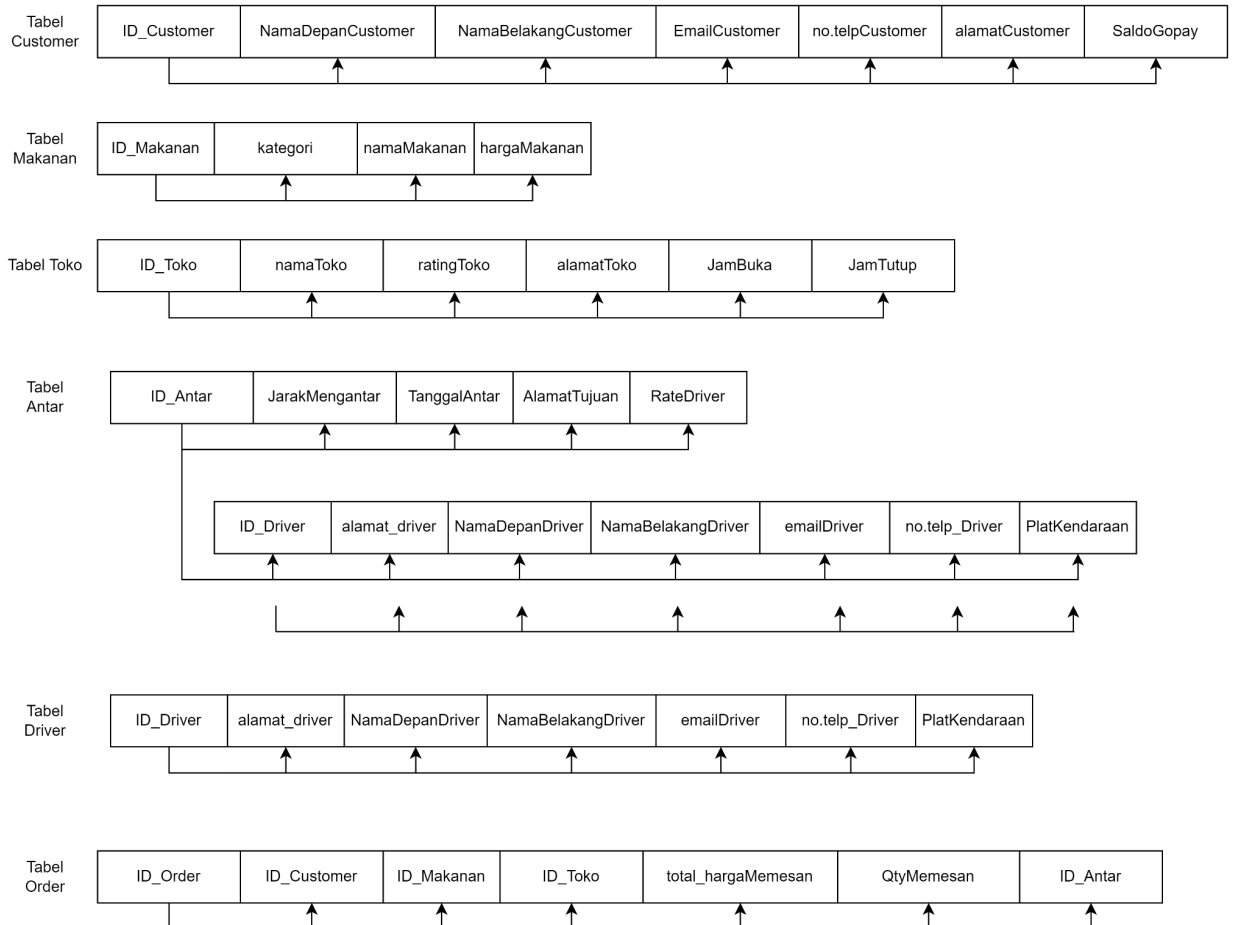
○ 2NF



Dikarenakan pada 1NF hanya terdapat 1 Primary Key, maka secara langsung semua kolom termasuk parsial dependent. Sehingga bentuk normalisasi 2NF sama dengan 1 NF.

ID\_Order → ID\_Antar, JarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, total\_hargaMemesan, QtyMemesan, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay, ID\_Makanan, kategori, namaMakanan, hargaMakanan, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan, ID\_Toko, namaToko, ratingToko, alamatToko, JamBuka, JamTutup

### ○ 3NF



- Karena namaDepanCustomer, namaBelakangCustomer, EmailCustomer, noTelpCustomer, AlamatCustomer, SaldoGopay bisa didapat dari ID\_Customer maka dibuat table baru untuk menghilangkan transitive table
- Karena kategori, namaMakanan, hargaMakanan bisa didapat dari ID\_Toko maka dibuat table baru untuk menghilangkan transitive table
- Karena jarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, noTelpDriver, platKendaraan bisa didapat dari ID\_Antar maka dibuat table baru untuk menghilangkan transitive table. Setelah itu karena, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, noTelpDriver, platKendaraan bisa didapat dari ID\_Driver maka dibuat table baru untuk menghilangkan transitive table pada table Antar
- Sisa dari table utama yaitu ID\_Order, ID\_Customer, ID\_Makanan, ID\_Toko, totalHarga, Qty, dan ID\_Antar menjadi table Order

ID\_Order → ID\_Customer, ID\_Makanan, ID\_Toko, ID\_Antar, ID\_Driver,  
total\_hargaMemesan, QtyMemesan

ID\_Customer → NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer,  
no.telpCustomer, alamatCustomer, SaldoGopay

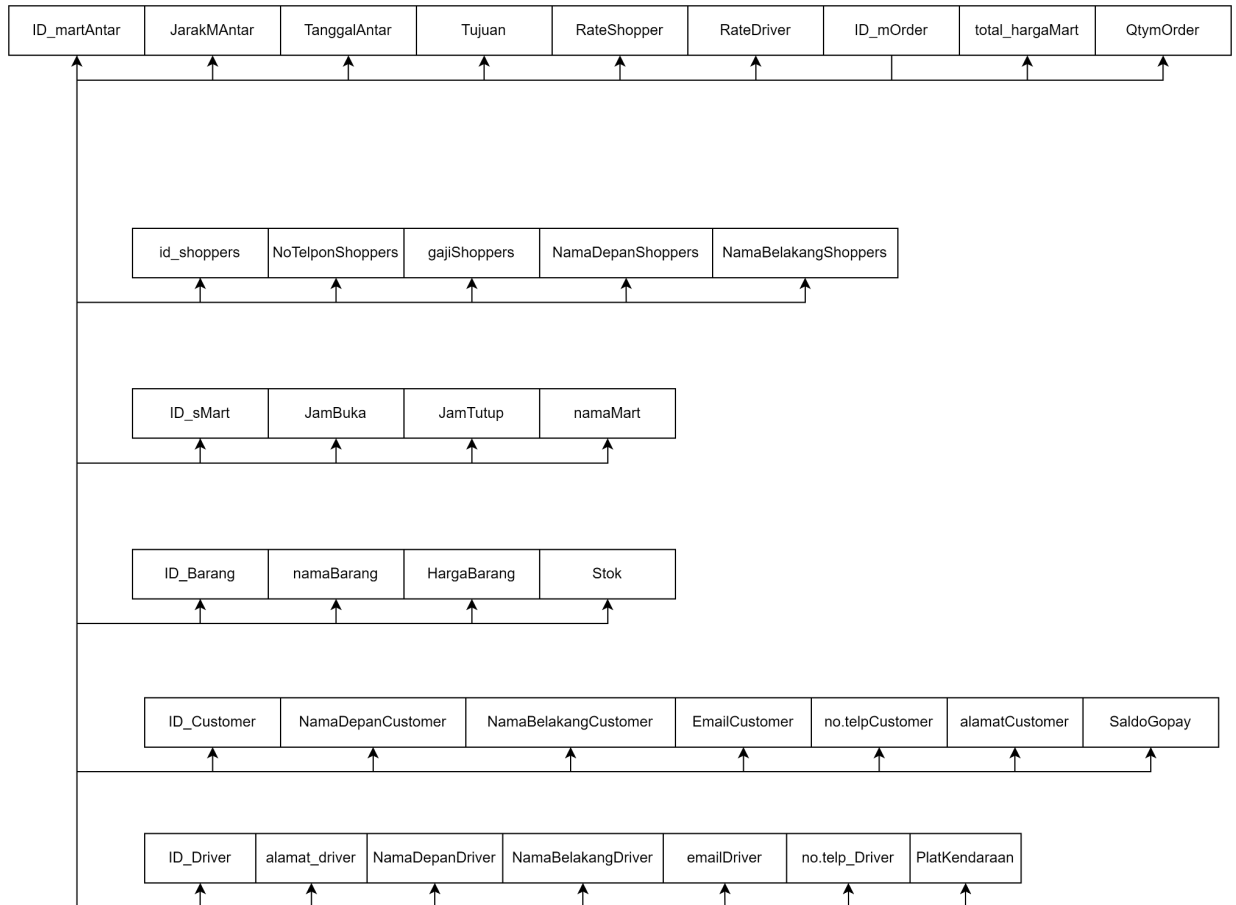
ID\_Makanan → ID\_Makanan, kategori, namaMakanan, hargaMakanan

ID\_Toko → namaToko, ratingToko, alamatToko, JamBuka, JamTutup

ID\_Antar → JarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, ID\_Customer,  
ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver,  
emailDriver, no.telp\_Driver, PlatKendaraan,

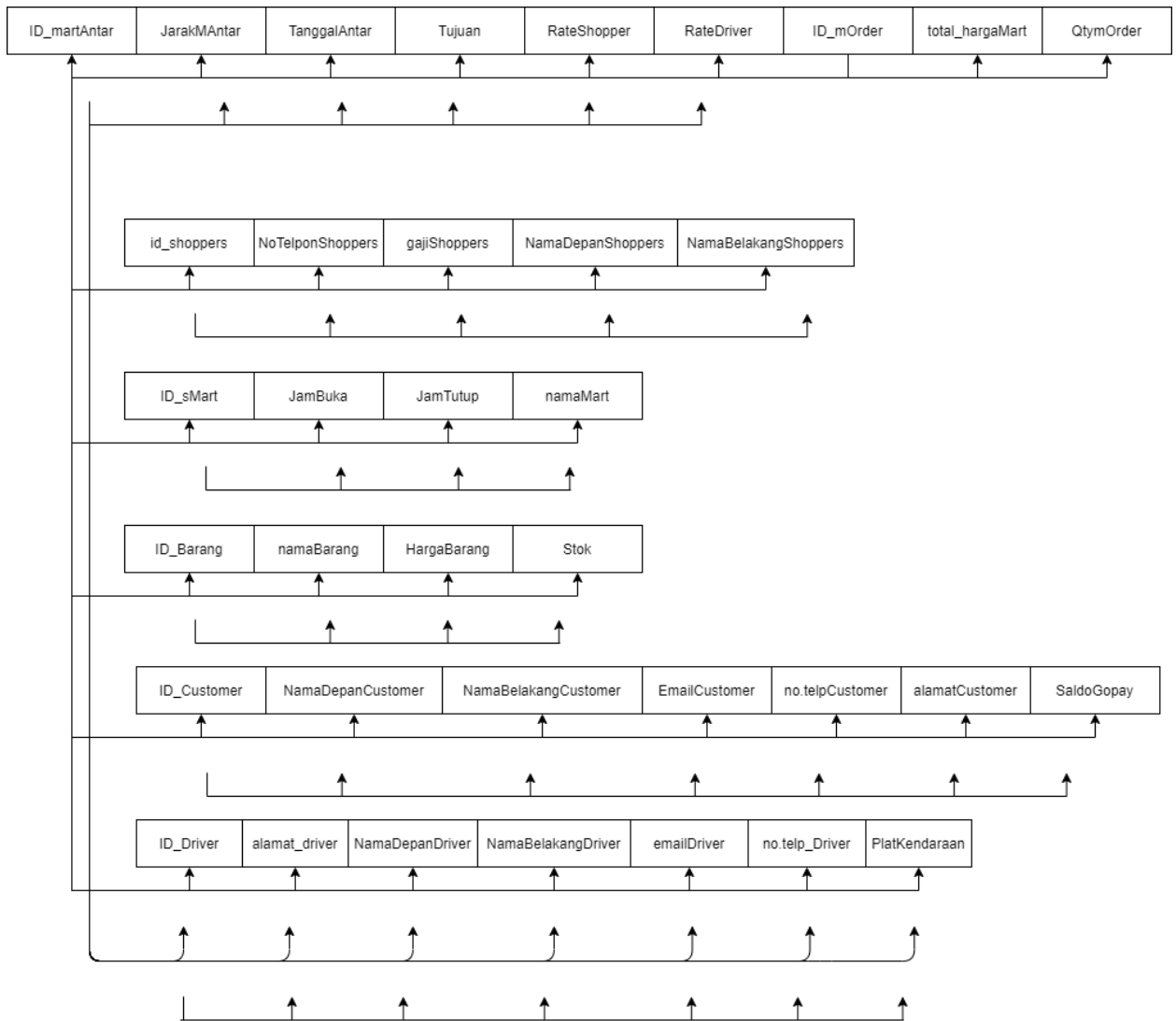
ID\_Driver → alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver,  
no.telp\_Driver, PlatKendaraan

- Normalisasi GoMart
  - Data Requirements



Dari data requirements tersebut yang menjadi Primary Key awal adalah ID\_mOrder

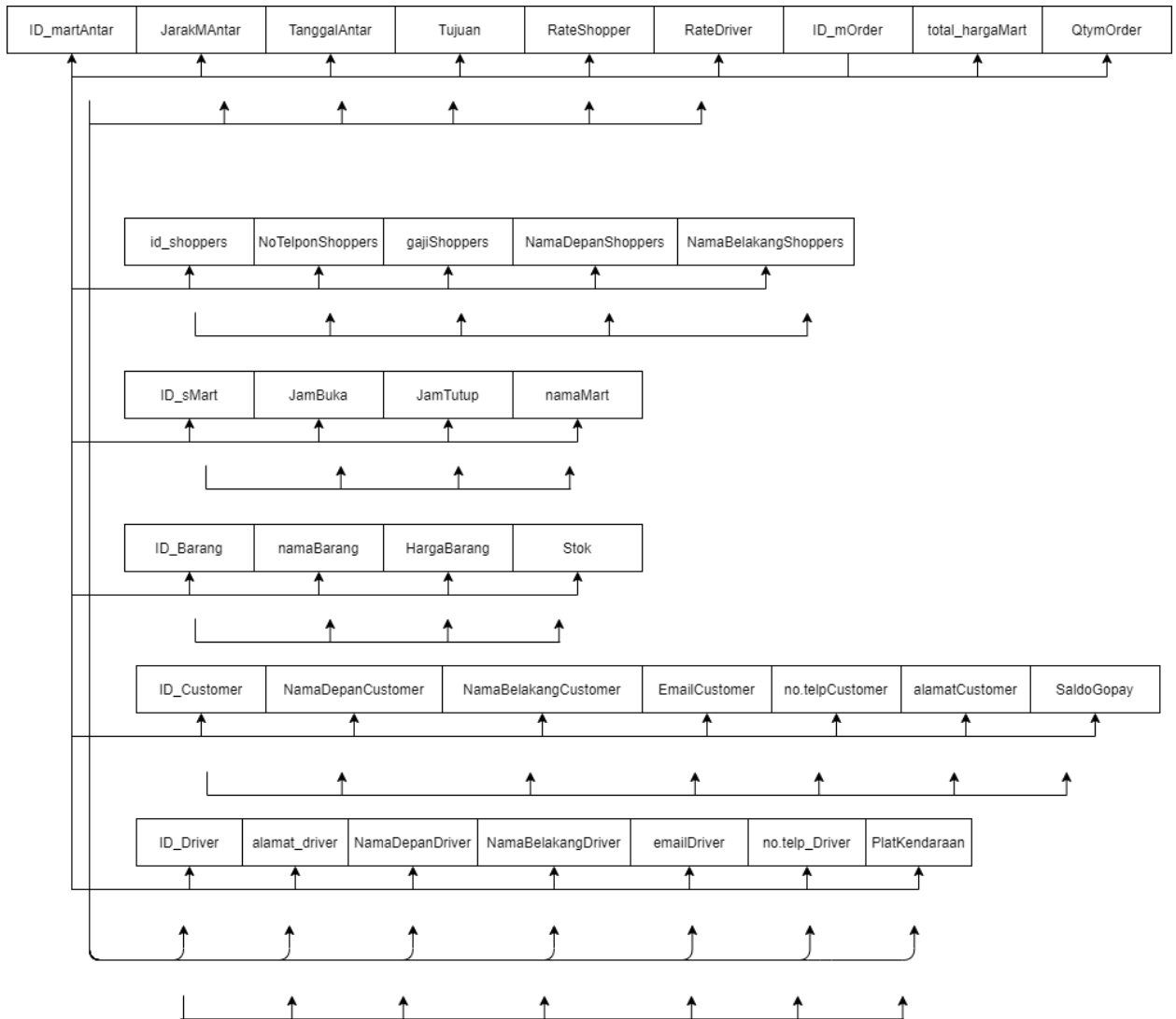
○ 1NF



Dari ID\_mOrder, tidak ada kolom yang bernilai multivalue maka dari itu tidak ada tambahan Primary Key.

ID\_mOrder → ID\_martAntar, JarakMAntar, TanggalAntar, Tujuan, RateShopper, RateDriver, ID\_mOrder, total\_hargaMart, QtymOrder, id\_shoppers, NoTelponShoppers, gajiShoppers, NamaDepanShoppers, NamaBelakangShoppers, ID\_sMart, JamBuka, JamTutup, namaMart, ID\_Barang, namaBarang, HargaBarang, Stok, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan

○ 2NF



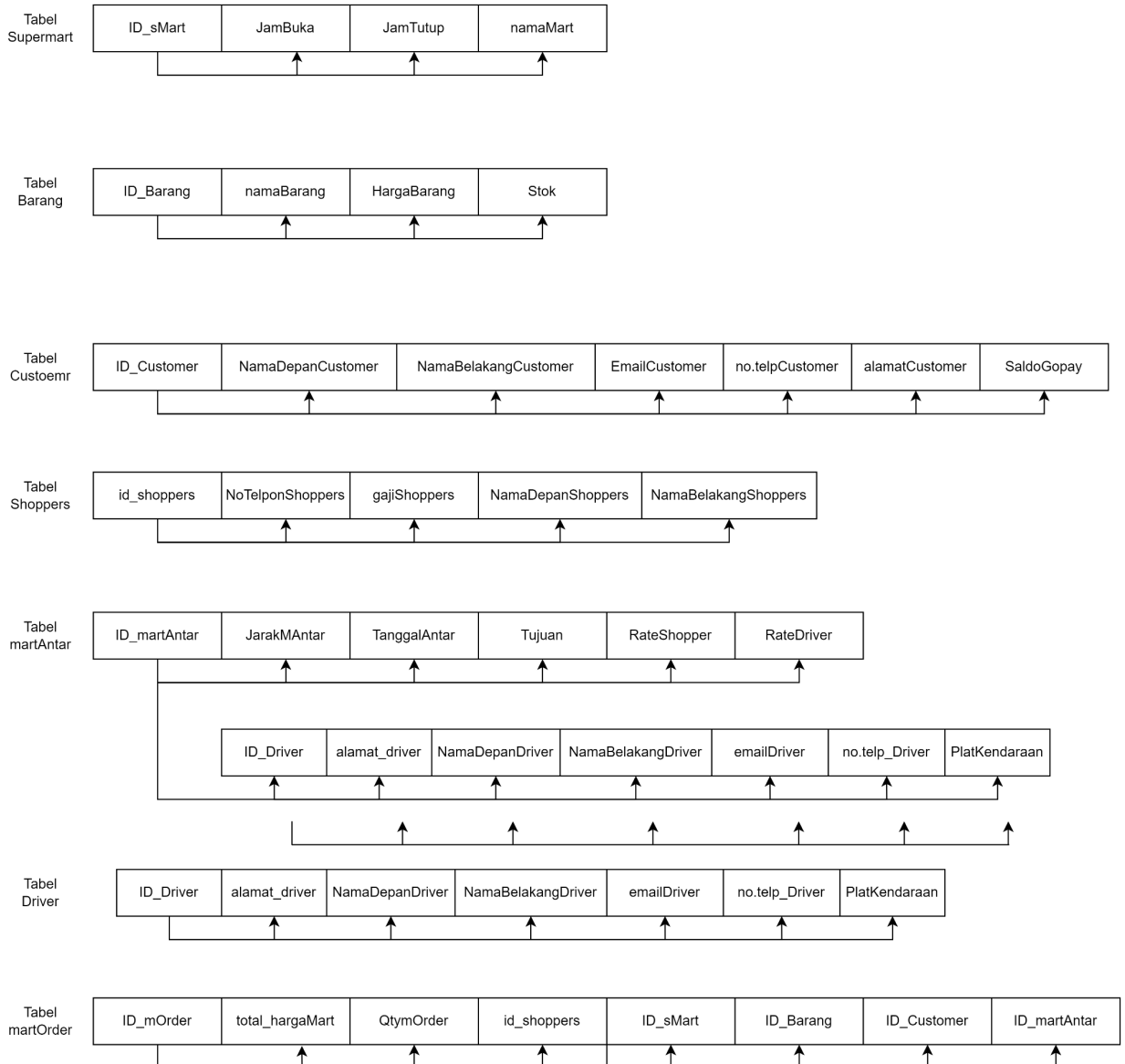
Dikarenakan pada 1NF hanya terdapat 1 Primary Key, maka secara langsung semua kolom termasuk parsial dependent. Sehingga bentuk normalisasi 2NF sama dengan 1 NF.

ID\_mOrder → ID\_martAntar, JarakMAntar, TanggalAntar, Tujuan, RateShopper, RateDriver, ID\_mOrder, total\_hargaMart, QtymOrder, id\_shoppers, NoTelponShoppers, gajiShoppers, NamaDepanShoppers, NamaBelakangShoppers, ID\_sMart, JamBuka, JamTutup, namaMart, ID\_Barang, namaBarang, HargaBarang, Stok, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay, ID\_Driver, alamat\_driver,

NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver,  
PlatKendaraan



### ○ 3NF



- Karena namaDepanCustomer, namaBelakangCustomer, EmailCustomer, noTelpCustomer, AlamatCustomer, SaldoGopay bisa didapat dari ID\_Customer maka dibuat table baru untuk menghilangkan transitive table
- Karena NamaDepanShoppers, NamaBelakangShoppers, NoTelponShoppers, dan gajiShoppers bisa didapat dari ID\_Shoppers maka dibuat table baru untuk menghilangkan transitive table
- Karena NamaMart, jamBuka, jamTutup bisa didapat dari ID\_sMart maka dibuat table baru untuk menghilangkan transitive table
- Karena NamaBarang, HargaBarang, stok bisa didapat dari ID\_Barang maka dibuat table baru untuk menghilangkan transitive table

- Karena jarakMengantar, TanggalAntar, AlamatTujuan, RateDriver, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, noTelpDriver, platKendaraan bisa didapat dari ID\_martAntar maka dibuat table baru untuk menghilangkan transitive table. Setelah itu karena, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, noTelpDriver, platKendaraan bisa didapat dari ID\_Driver maka dibuat table baru untuk menghilangkan transitive table pada table mAntar
- Sisa dari table utama yaitu ID\_mOrder, ID\_Customer, ID\_sMart, ID\_Barang, ID\_Shoppers, total\_HargaMart, Qty, dan ID\_martAntar menjadi table martOrder

ID\_sMart → JamBuka, JamTutup, namaMart

ID\_Barang → namaBarang, HargaBarang, Stok

ID\_Customer → NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay

Id\_shoppers → NoTelponShoppers, gajiShoppers, NamaDepanShoppers, NamaBelakangShoppers

ID\_martAntar → JarakMAntar, TanggalAntar, Tujuan, RateShopper, RateDriver, ID\_mOrder, ID\_Driver, alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan

ID\_Driver → alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, no.telp\_Driver, PlatKendaraan

ID\_mOrder → ID\_martAntar, total\_hargaMart, QtymOrder, Id\_shoppers, ID\_sMart, ID\_Barang, ID\_Customer

- Normalisasi Manajemen Admin-Customer

|          |                |                   |            |               |           |           |             |                   |                      |               |                 |                |            |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-------------|-------------------|----------------------|---------------|-----------------|----------------|------------|
| ID_Admin | NamaDepanAdmin | NamaBelakangAdmin | EmailAdmin | NoTelponAdmin | UmurAdmin | GajiAdmin | ID_Customer | NamaDepanCustomer | NamaBelakangCustomer | EmailCustomer | no.telpCustomer | alamatCustomer | SaldoGopay |
|          | ↑              | ↑                 | ↑          | ↑             | ↑         | ↑         | ↑           | ↑                 | ↑                    | ↑             | ↑               | ↑              | ↑          |

- Data Requirements

Dari data requirements tersebut yang menjadi Primary Key awal adalah ID\_Admin

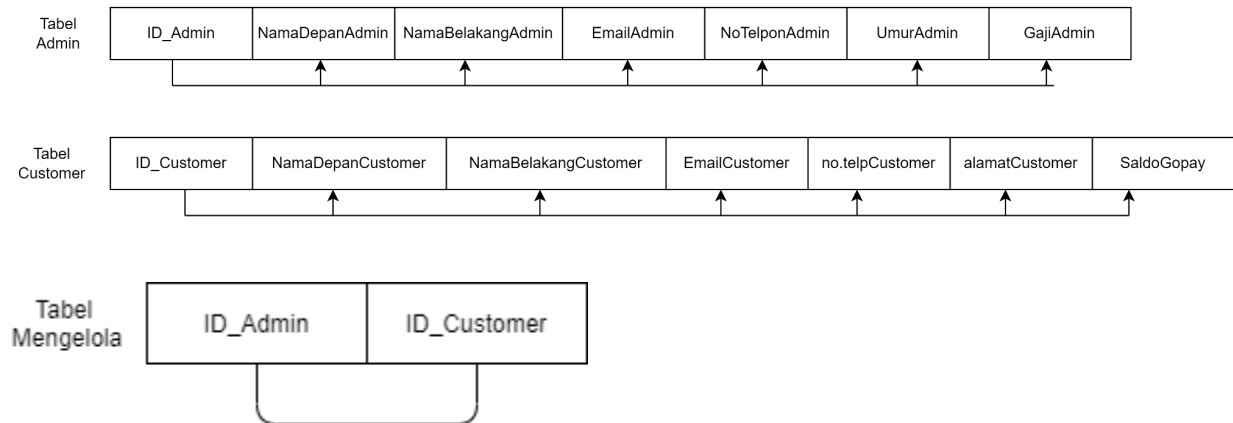
|          |                |                   |            |               |           |           |             |                   |                      |               |                 |                |            |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-------------|-------------------|----------------------|---------------|-----------------|----------------|------------|
| ID_Admin | NamaDepanAdmin | NamaBelakangAdmin | EmailAdmin | NoTelponAdmin | UmurAdmin | GajiAdmin | ID_Customer | NamaDepanCustomer | NamaBelakangCustomer | EmailCustomer | no.telpCustomer | alamatCustomer | SaldoGopay |
|          | ↑              | ↑                 | ↑          | ↑             | ↑         | ↑         |             | ↑                 | ↑                    | ↑             | ↑               | ↑              | ↑          |
|          | ↑              | ↑                 | ↑          | ↑             | ↑         | ↑         |             | ↑                 | ↑                    | ↑             | ↑               | ↑              | ↑          |

- 1NF

Karena terdapat multivalued pada kolom ID\_Customer dari ID\_Admin, maka ID\_Customer menjadi primary key untuk menghilangkan multivalued

ID\_Admin → NamaDepanAdmin, NamaBelakangAdmin, EmailAdmin, NoTelponAdmin, UmurAdmin, GajiAdmin, ID\_Customer, NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay

○ 2NF



- Karena untuk mengetahui NamaDepanAdmin, NamaBelakangAdmin, EmailAdmin, noTelponAdmin, UmurAdmin, GajiAdmin hanya dari ID\_Admin, maka dibuat table baru untuk menghilangkan parsial dependency
- Karena untuk mengetahui namaDepanCustomer, namaBelakangCustomer, EmailCustomer, noTelpCustomer, AlamatCustomer, SaldoGopay bisa didapat dari ID\_Customer maka dibuat table baru untuk menghilangkan parsial dependency
- Sisa table utama adalah ID\_Admin dan ID\_Customer menjadi table mengelola

ID\_Admin → NamaDepanAdmin, NamaBelakangAdmin, EmailAdmin, NoTelponAdmin, UmurAdmin, GajiAdmin

ID\_Customer → NamaDepanCustomer, NamaBelakangCustomer, EmailCustomer, no.telpCustomer, alamatCustomer, SaldoGopay

ID\_Admin, ID\_Customer → -

- Normalisasi Manajemen Admin-Driver
  - Data Requirements

|          |                |                   |            |               |           |           |           |               |                 |                    |             |                |               |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-----------|---------------|-----------------|--------------------|-------------|----------------|---------------|
| ID_Admin | NamaDepanAdmin | NamaBelakangAdmin | EmailAdmin | NoTelponAdmin | UmurAdmin | GajiAdmin | ID_Driver | alamat_driver | NamaDepanDriver | NamaBelakangDriver | emailDriver | no.telp_Driver | PlatKendaraan |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-----------|---------------|-----------------|--------------------|-------------|----------------|---------------|

Dari data requirements tersebut yang menjadi Primary Key awal adalah ID\_Admin

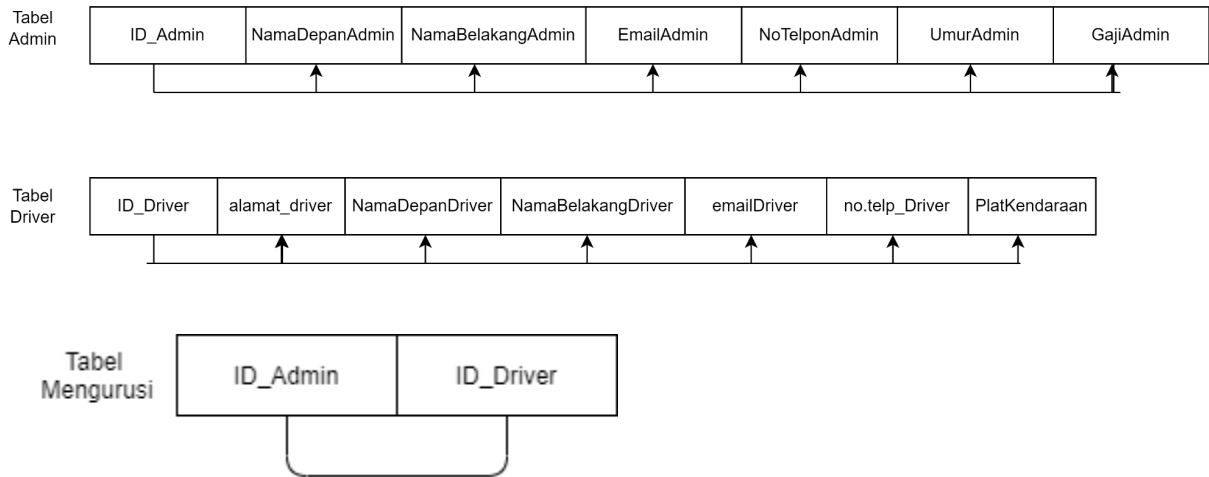
- 1NF

|          |                |                   |            |               |           |           |           |               |                 |                    |             |                |               |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-----------|---------------|-----------------|--------------------|-------------|----------------|---------------|
| ID_Admin | NamaDepanAdmin | NamaBelakangAdmin | EmailAdmin | NoTelponAdmin | UmurAdmin | GajiAdmin | ID_Driver | alamat_driver | NamaDepanDriver | NamaBelakangDriver | emailDriver | no.telp_Driver | PlatKendaraan |
|----------|----------------|-------------------|------------|---------------|-----------|-----------|-----------|---------------|-----------------|--------------------|-------------|----------------|---------------|

Karena terdapat multivalued pada kolom ID\_Driver dari ID\_Admin, maka ID\_Driver menjadi primary key untuk menghilangkan multivalued

ID\_Admin → NamaDepanAdmin NamaBelakangAdmin EmailAdmin NoTelponAdmin  
 UmurAdmin GajiAdmin ID\_Driver alamat\_driver NamaDepanDriver  
 NamaBelakangDriver emailDriver no.telp\_Driver PlatKendaraan

○ 2NF



- Karena untuk mengetahui NamaDepanAdmin, NamaBelakangAdmin, EmailAdmin, noTelponAdmin, UmurAdmin, GajiAdmin hanya dari ID\_Admin, maka dibuat table baru untuk menghilangkan parsial dependency
- Karena untuk mengetahui alamat\_driver, NamaDepanDriver, NamaBelakangDriver, emailDriver, noTelpDriver, platKendaraan bisa didapat dari ID\_Driver, maka dibuat table baru untuk parsial dependency
- Sisa table utama adalah ID\_Admin dan ID\_Driver menjadi table mengurus

ID\_Admin → NamaDepanAdmin NamaBelakangAdmin EmailAdmin NoTelponAdmin  
UmurAdmin GajiAdmin

ID\_Driver → alamat\_driver NamaDepanDriver NamaBelakangDriver emailDriver no.telp\_Driver  
PlatKendaraan

ID\_Admin, ID\_Driver → -

Penjelasan Tambahan:

- Normalisasi untuk konteks GoFood dan GoMart terpisah karena akan bentrok atau berebutan entitas customer dan driver karena kedua konteks membutuhkan 2 entitas tersebut
- Entitas Admin dibuat terpisah dari konteks GoFood dan GoMart karena Admin tidak ada sangkut pautnya dengan konteks GoFood dan GoMart, walaupun ada korelasi dengan customer dan driver, akan terjadi bentrok atau berebutan pada customer dan driver
- Normalisasi konteks manajemen terbagi menjadi 2 yaitu admin-customer dan admin-driver. Dibuat terpisah karena apabila digabung, ID\_Customer dan ID\_Driver akan menjadi kolom multivalued yg mana nantinya malah akan membuat kolom menjadi double dari seharusnya (jika dilihat dari excel). Jika hanya 1 multivalued, misalkan hanya ID\_Customer tetapi ID\_Driver tidak, maka kelanjutannya akan membuat ID\_Driver menjadi tidak primary key dan menjadi parsial untuk ID\_Admin, yang mana seharusnya kedua ID\_Driver dan ID\_Admin menjadi primary key.

| ID_Admin | ID_Customer  | ID_Driver    |
|----------|--------------|--------------|
| A0001    | C0001, C0002 | D0001, D0002 |
|          |              |              |
| ID_Admin | ID_Customer  | ID_Driver    |
| A0001    | C0001        | D0001,D0002  |
| A0001    | C0002        | D0001,D0002  |
|          |              |              |
| ID_Admin | ID_Customer  | ID_Driver    |
| A0001    | C0001        | D0001        |
| A0001    | C0001        | D0002        |
| A0001    | C0002        | D0001        |
| A0001    | C0002        | D0002        |