IF2040 Pemodelan Basis Data  
Diagram Relasional dan Implementasi Basis Data

**ShopeeFood**

****

**Disusun oleh:**

**Kelompok K3-G05**

Anggita Najmi Layali / 18223122

Muhammad Dzaky Atha F / 18223124

M Rabbani K A / 18223130

Sonya Putri Fadilah / 18223138

**Program Studi Sistem dan Teknologi Informasi**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2024**

# Daftar Isi

[**Daftar Isi 2**](#_56jc5bfx9db8)

[**1. Deskripsi Sistem 3**](#_q639d0gvm8gm)

[**2. ER Diagram 5**](#_vvp7t4kcbgcv)

[**3. Pemetaan ER Diagram menjadi Model Relasional 7**](#_7n3cght4qkaq)

[3.1. Pemetaan entity menjadi relasi 7](#_s7olztfgja0l)

[3.2. Pemetaan relationship menjadi relasi 8](#_rr9zf3bficu1)

[3.3. Pemetaan specialization/generalization menjadi relasi 10](#_3253sfysd6b7)

[3.4. Pemetaan Multivalued Attributes 10](#_x8s6hstwm9u7)

[3.5. Derived Attributes 11](#_nvdegn9uf35f)

[3.6. Foreign Keys 11](#_jtojuvlerp3x)

[3.7. Hasil Pemetaan 12](#_5521g22acv99)

[**4. Diagram Model Relasional 13**](#_7edmp4q5oj00)

[**5. Daftar Relasi 14**](#_weo34rs9v0p1)

[5.1. Relasi User 14](#_svqmh91io989)

[5.2. Relasi User\_telp 15](#_qjxfip5yaf0b)

[5.3. Relasi Shopeepay 16](#_ysdu0grysg7w)

[5.4. Relasi Transaction 17](#_yhj4or6fdhsa)

[5.5. Relasi Category 18](#_76iyg2lqmaop)

[5.6. Relasi Restaurant 18](#_9gsugd116y6p)

[5.7. Relasi Menu 19](#_r8gcit4kq8rm)

[5.8. Relasi Pesanan 21](#_4q6fxgkhg0wj)

[5.9. Relasi Order Item 21](#_juax24nqsuk0)

[5.10. Relasi Order Address 22](#_ra1tduch30ou)

[5.11. Relasi Driver 23](#_awj701wx6mlv)

[5.12. Relasi History 24](#_64xz71lti89i)

[5.13. Relasi Refund 25](#_am6iovqmvha)

[**Daftar Pustaka 26**](#_kqn8oj5358fn)

[**Pembagian Tugas 27**](#_akibheg7liwu)

[**Lampiran 28**](#_ygkxy47fy3pp)

[Moments of Meeting 28](#_t5kvsih7o68k)

# Deskripsi Sistem



*Gambar 1. Logo ShopeeFood*

*ShopeeFood* adalah salah satu fitur dalam aplikasi *online* *shop Shopee* yang menyediakan jasa layanan pesan antar makanan. Dengan *ShopeeFood*, pengguna dapat memesan makanan secara online. Kelompok kami tertarik untuk menganalisisnya karena selain menjadi pengguna, *ShopeeFood* juga menjadi salah satu pilihan banyak orang saat memesan makanan secara *online* meskipun tidak sepopuler aplikasi pemesanan makanan lain, seperti *GoFood*. Namun, kami sebagai mahasiswa Sistem dan Teknologi Informasi (STI) ITB yang sedang mempelajari mata kuliah pemodelan basis data (PBD) tertarik untuk menganalisis sistem yang digunakan *ShopeeFood* dalam menyediakan jasa layanan antar makanan. Basis data yang akan kami analisis adalah data *user*, data menu makanan, data kategori makanan, data restoran, data *Shopeepay* sebagai e-*wallet* bawaan Shopee, data *driver*, dan data pemesanan beserta riwayatnya.

Untuk mengakses *ShopeeFood*, *user* perlu membuka aplikasi *Shopee* dan *user* akan diberikan dua opsi yaitu fitur “*log in*” jika sudah mendaftar dan fitur “*create account*” jika belum memiliki akun *Shopee*. Setiap user akan memiliki ID yang unik, nama lengkap, tanggal lahir, jenis kelamin, alamat, nomor *handphone*, dan email. Kemudian, pada halaman beranda Shopee terdapat fitur *ShopeeFood* yang dapat langsung diakses dengan meng-klik ikonnya saja. Setelah itu, *user* akan berada di halaman beranda *ShopeeFood* dan dapat mencari makanan yang ingin dipesan.

Saat ingin mencari makanan, *user* dapat memilih melalui kategori yang tertera pada halaman beranda. Setiap kategori makanan memiliki ID kategori yang unik dan nama kategori. Setelah memilih kategori, terdapat beberapa restoran yang *user* bisa pilih. Setiap restoran memiliki ID restoran yang unik, nama restoran, alamat, dan rating restoran (1-5) yang dikalkulasi dari rata-rata rating restoran oleh setiap *user* setiap kali melakukan pemesanan pada restoran tersebut. Setiap kategori terdapat beberapa restoran dan setiap restoran bisa termasuk dalam beberapa kategori.

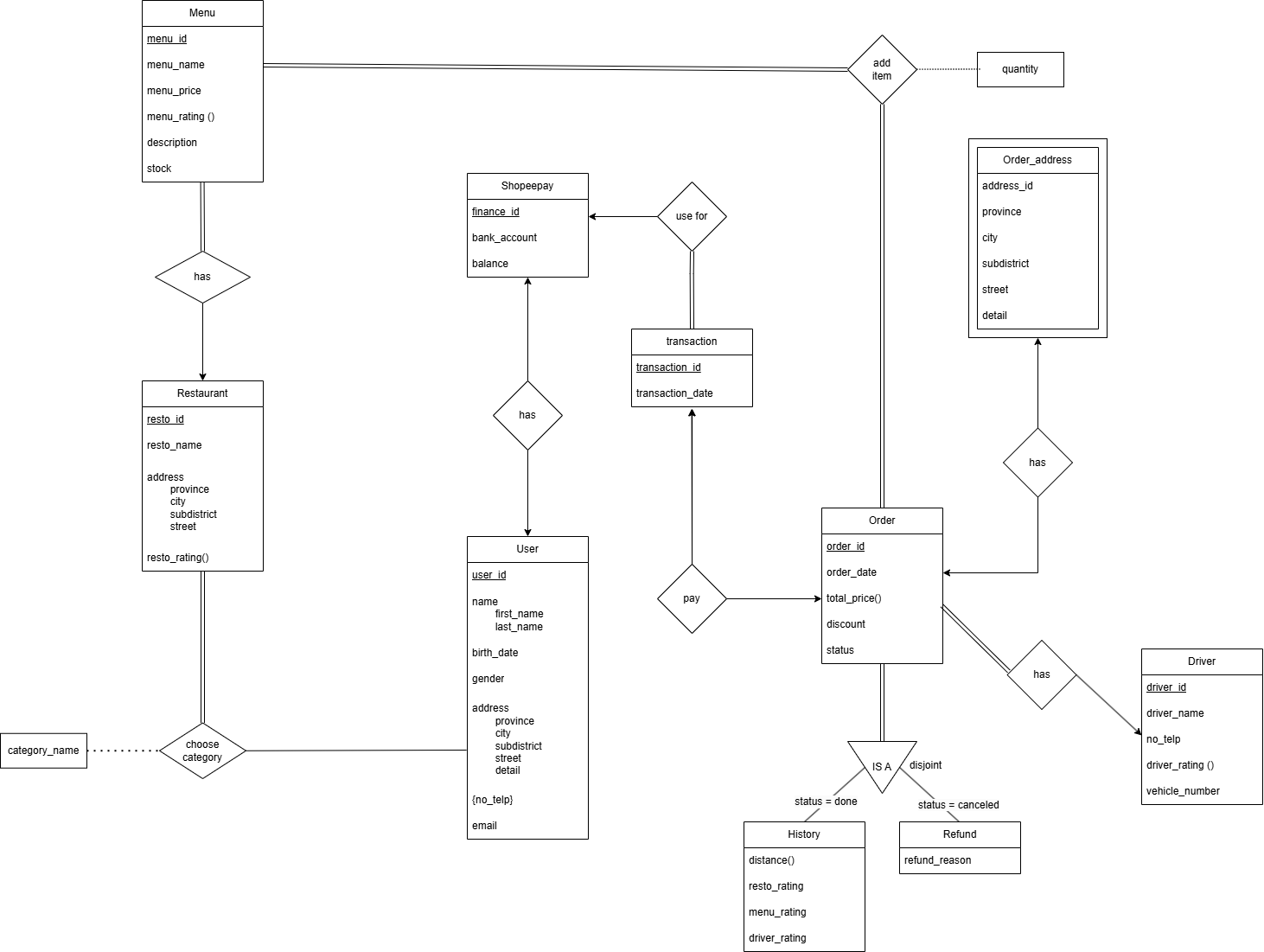
*User* juga dapat mencari menu makanan yang tersedia baik dengan langsung mencari nama makanan tersebut, maupun melihat-lihat melalui restoran. Setiap menu memiliki ID menu yang unik, nama menu, rating menu (1-5) yang dikalkulasi dari rata-rata rating menu oleh seluruh *user* tiap kali melakukan pemesanan menu tersebut, deskripsi menu, dan stok yang tersedia.

Setiap *user* dapat melakukan lebih dari satu pemesanan. Setiap pesanan akan dicatat ID pesanan yang unik, tanggal pemesanan, total harga yang dikalkulasikan dari harga tiap makanan yang dipesan, diskon, dan status pemesanan yang menyatakan apakah pesanan yang dilakukan selesai atau dibatalkan. Setiap pesanan yang selesai dilakukan akan masuk ke dalam riwayat pesanan. Setiap pesanan yang masuk ke riwayat akan memiliki jarak–dikalkulasikan dari jarak antara lokasi pemesanan user dengan lokasi restoran makanan yang dipesan, dan rating yang terdiri dari rating restoran, menu, dan driver. Namun, jika pemesanan yang dilakukan dibatalkan, maka akan masuk ke bagian pengembalian dana. Pada pengembalian dana, terdapat alasan kenapa pesanan tersebut dibatalkan.

Untuk setiap pesanan yang dilakukan akan dicatat alamat yang menjadi lokasi pemesanan *user* saat itu. Lokasi pemesanan dapat berubah, tanpa merubah alamat yang telah disimpan pada profil *user*. Setiap pesanan dapat terdiri dari beberapa menu makanan dengan jumlahnya masing-masing. Setiap menu yang dipesan memiliki jumlah dan harga per itemnya. Selain itu, setiap pesanan akan diantar oleh seorang driver. Setiap driver memiliki ID yang unik, nama, nomor telepon, rating (1-5) yang dikalkulasi dari rata-rata rating oleh setiap *use*r tiap kali pesanan makanan diantar oleh driver tersebut, dan nomor kendaraan.

Setiap pemesanan yang dilakukan oleh *user* harus diselesaikan pembayarannya. Setiap transaksi pembayaran akan menggunakan *Shopeepay* dan akan dicatat ID transaksi yang unik, tanggal, dan juga harga akhir yang dihitung berdasarkan harga total.

# ER Diagram



*Gambar 2. ER Diagram*

Penjelasan ER Diagram

Asumsi:

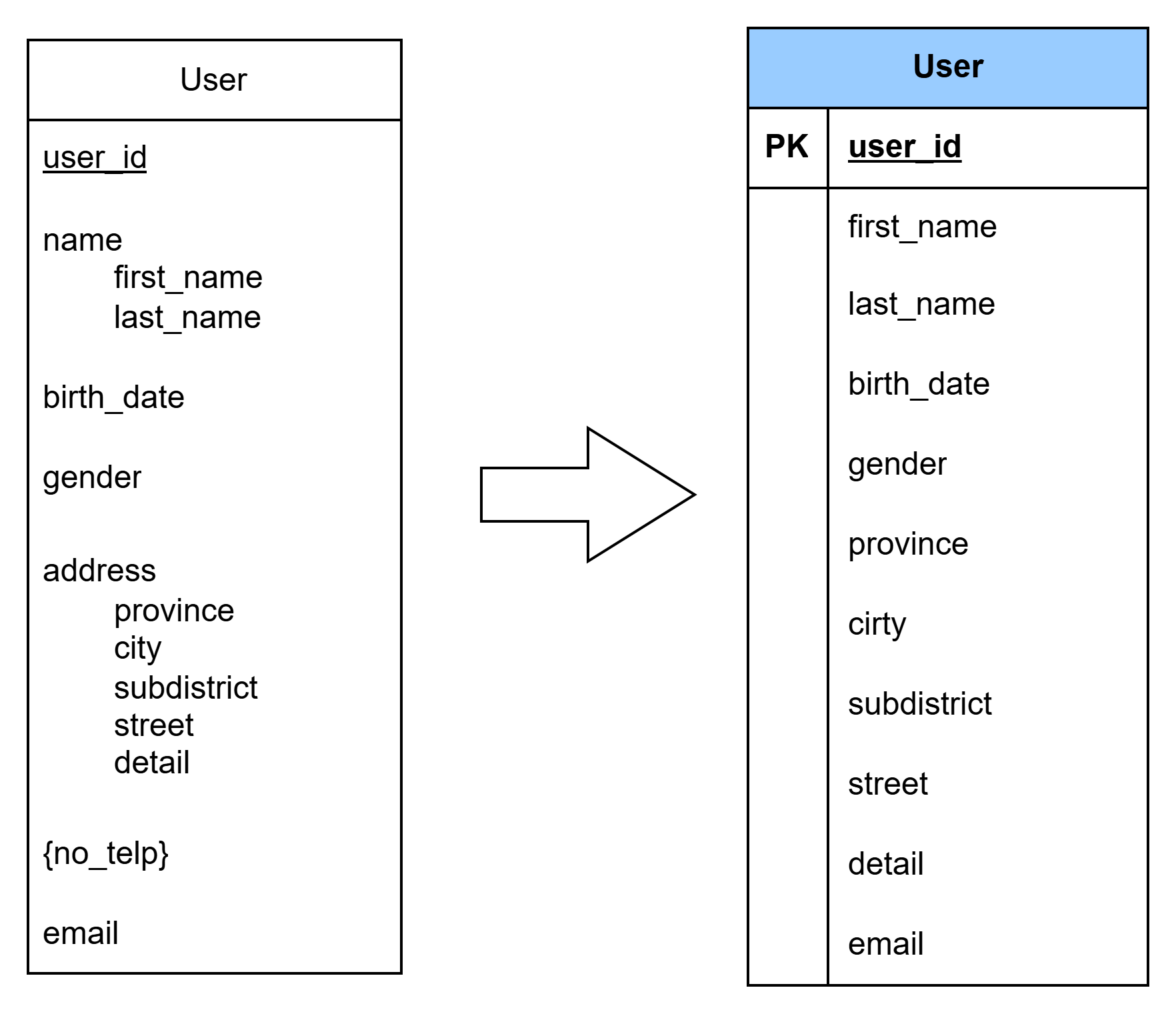
1. POV *user* sebagai *customer*/pembeli (bukan *owner merchant*/resto).
2. Setiap *user* hanya memiliki satu akun *Shopeepay* dan setiap *transaction* hanya melalui *Shopeepay*
3. Setiap pemesanan yang dilakukan sudah dibayar.
4. Setiap restoran dan menunya dapat memiliki nama yang sama. Oleh karena itu, restoran dan menu dibedakan dengan resto\_id dan menu\_id.
5. *Customer* dapat memesan beberapa menu makanan dalam satu kali Pesanan dan memillih jumlahnya masing-masing.
6. *Rating* dapat dilakukan apabila pemesanan dalam status selesai dengan satu pemesanan satu kali *rating*, *customer* juga tidak harus memberikan *rating*. *Rating* yang berikan adalah *rating* *restaurant*, menu, dan *driver*.
7. Entitas *category* hanya menyatakan jenis atau tipe menu di restoran secara umum, misalnya ayam, roti, kopi dan lain sebagainya.
8. Setiap restoran memiliki minimal 1 kategori dan dapat memiliki >1 kategori.
9. Setiap *customer* harus memilih minimal 1 menu sebelum melakukan pemesanan.

Hal yang tidak dapat ditulis dalam ER:

1. Dalam entity *Pesanan*, atribut *discount* maksudnya adalah “*amount discount*”-nya, sehingga bernilai angka. Misalkan *discount* = 10, berarti diskon 10%, jika tidak ada maka akan bernilai discount = 0.
2. *Driver\_rating* pada entity “Driver”, *resto\_rating* dari entity “Restaurant”, dan *menu\_rating* dari entity “Menu” adalah hasil kalkulasi rata-rata dari masing-masing rating yang diberikan di entity “history”.
3. Rating berskala 1-5.

# Pemetaan ER Diagram menjadi Model Relasional

## Pemetaan *entity* menjadi relasi



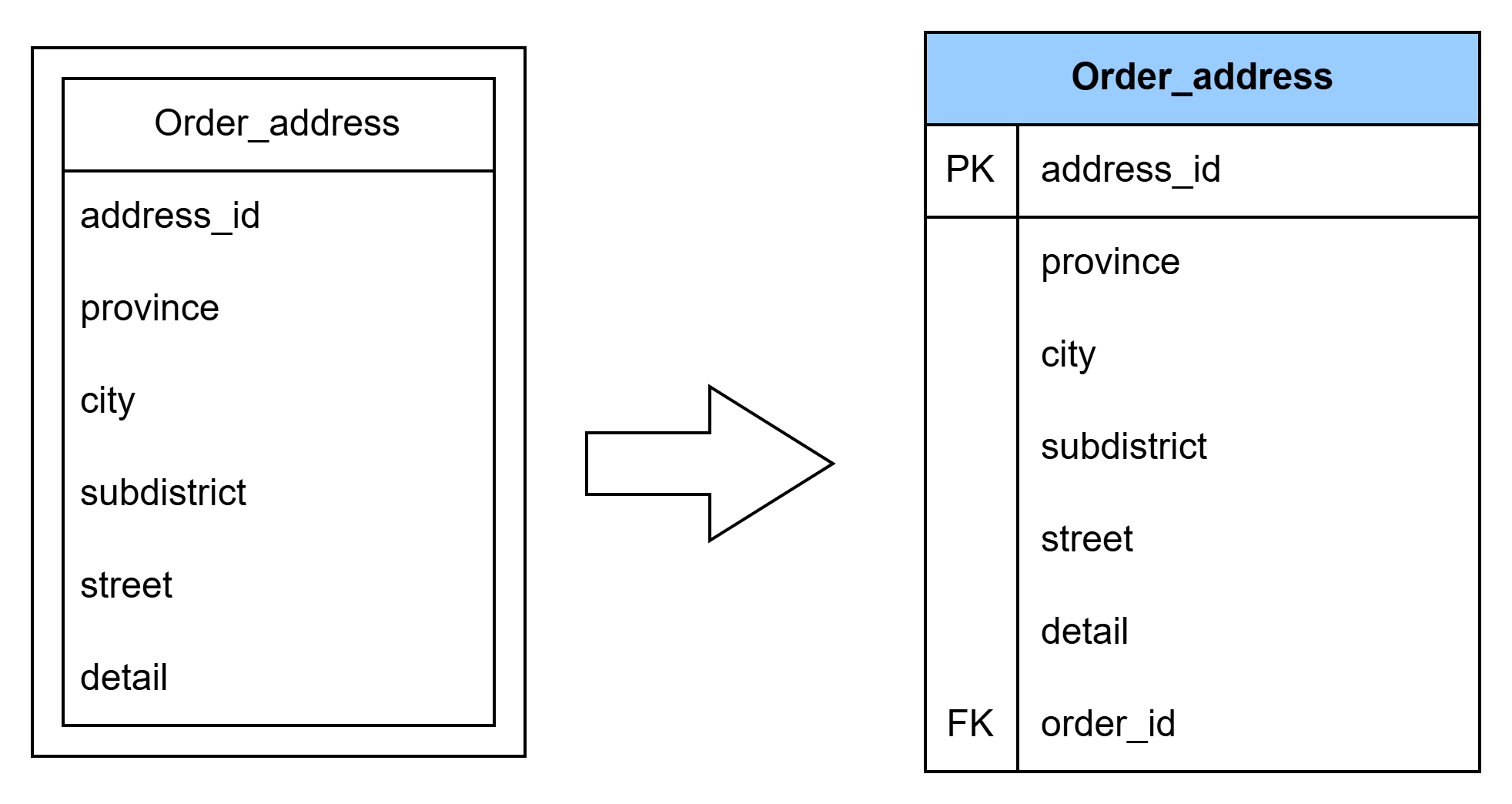
***Strong Entity***

Untuk memetakan *strong entity* menjadi suatu relasi, dibuat tabel dengan atribut yang sama dan mengidentifikasi *primary key* serta *foreign key*-nya. Terdapat perlakuan khusus untuk beberapa atribut tertentu:

* *Composite attributes* dihapus dan langsung mencantumkan isinya sebagai atribut.
* *Derived attribute*s dihapus, nantinya akan langsung dibuat view pada SQL.
* *Multivalued attributes* dibuat relasi baru.

| * 1. ***user*** *= (user\_id, first\_name, last\_name, birth\_date, gender, province, city, subdistrict, street, detail, email)*   2. ***restaurant*** *= (resto\_id, resto\_name, province, city, subdistrict, street, detail)*   3. ***menu*** *= (menu\_id, menu\_name, menu\_price, description, stock)*   4. ***pesanan*** *= (order\_id, order\_date, discount, status)*   5. ***transaction*** *= (transaction\_id, transaction\_date)*   6. ***shopeepay*** *= (finance\_id, bank\_account, balance)*   7. ***driver*** *= (driver\_id, driver\_name, no\_telp, vehicle\_number)* |
| --- |

***Weak Entity***

******

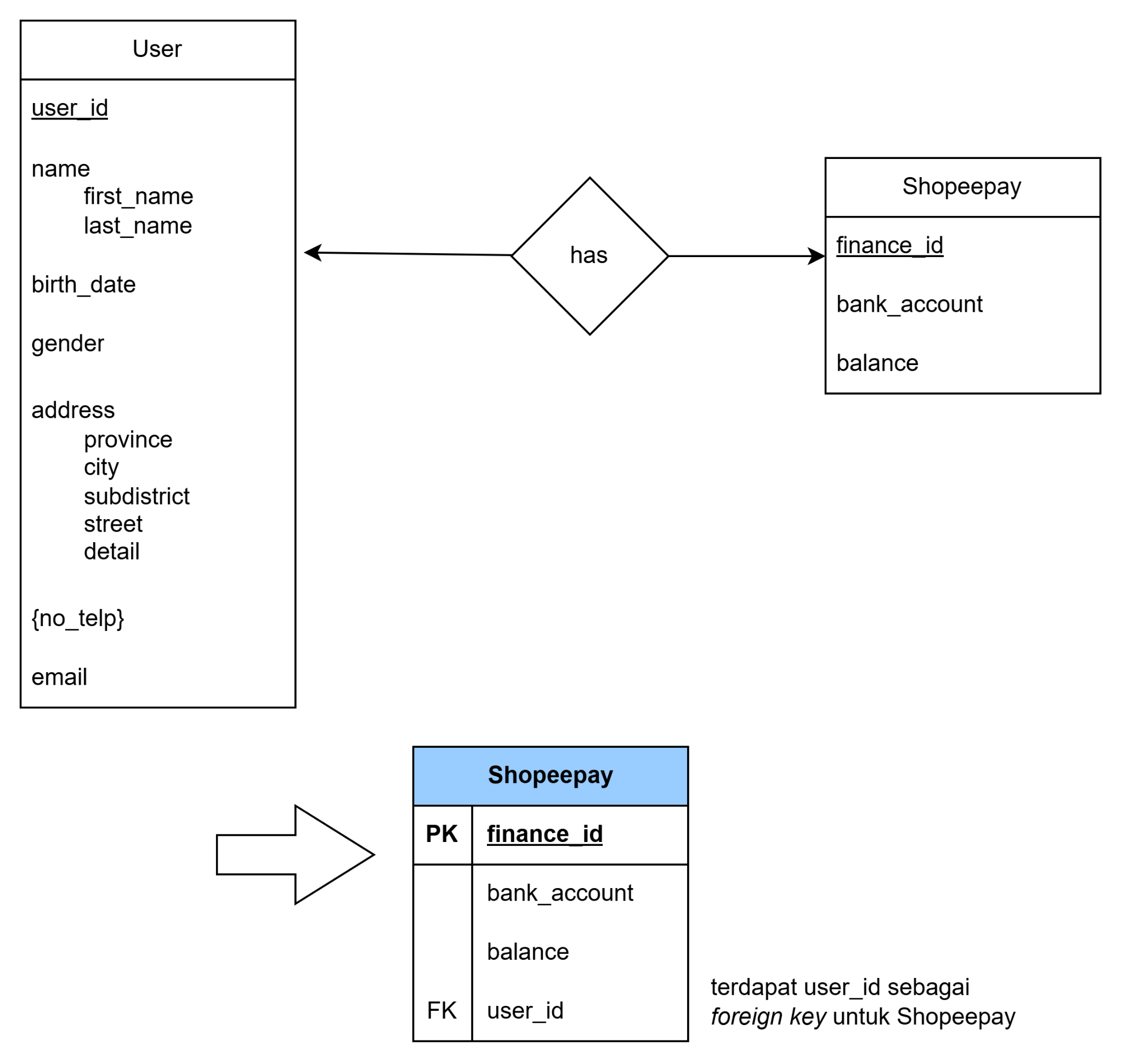
Pemetaan *weak entity* akan memiliki *foreign key* yang berasal dari *primary key* *strong entity*-nya dan juga akan memiliki sebuah *discriminator* yang menjadi *primary key*.

| ***order\_address*** *= (address\_id, province, city, subdistrict, street, detail, order\_id)* |
| --- |

## Pemetaan *relationship* menjadi relasi

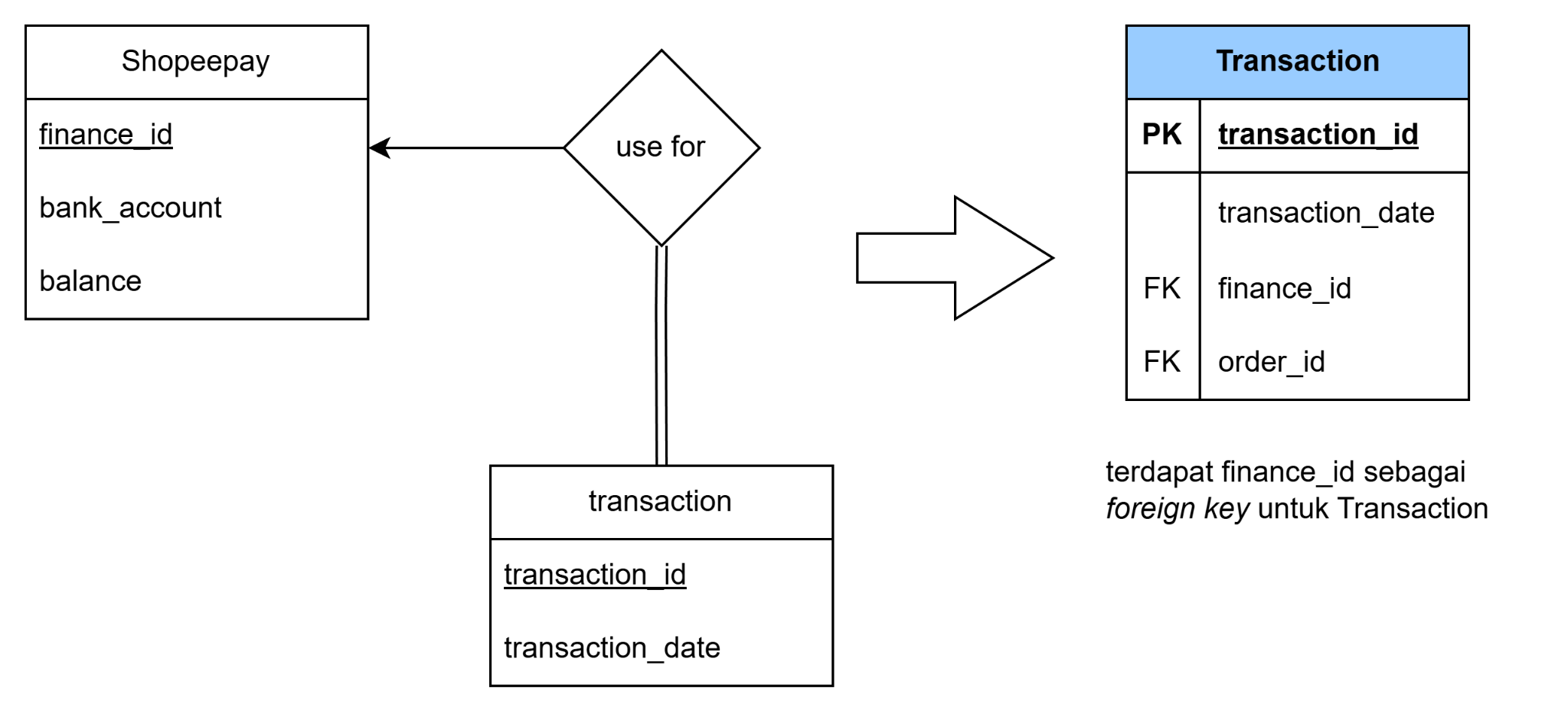
Dalam memetakan relationship menjadi atribut, terdapat beberapa perlakuan tergantung pada tipe *relationship*-nya.

1. **One-to-one relationship**

****

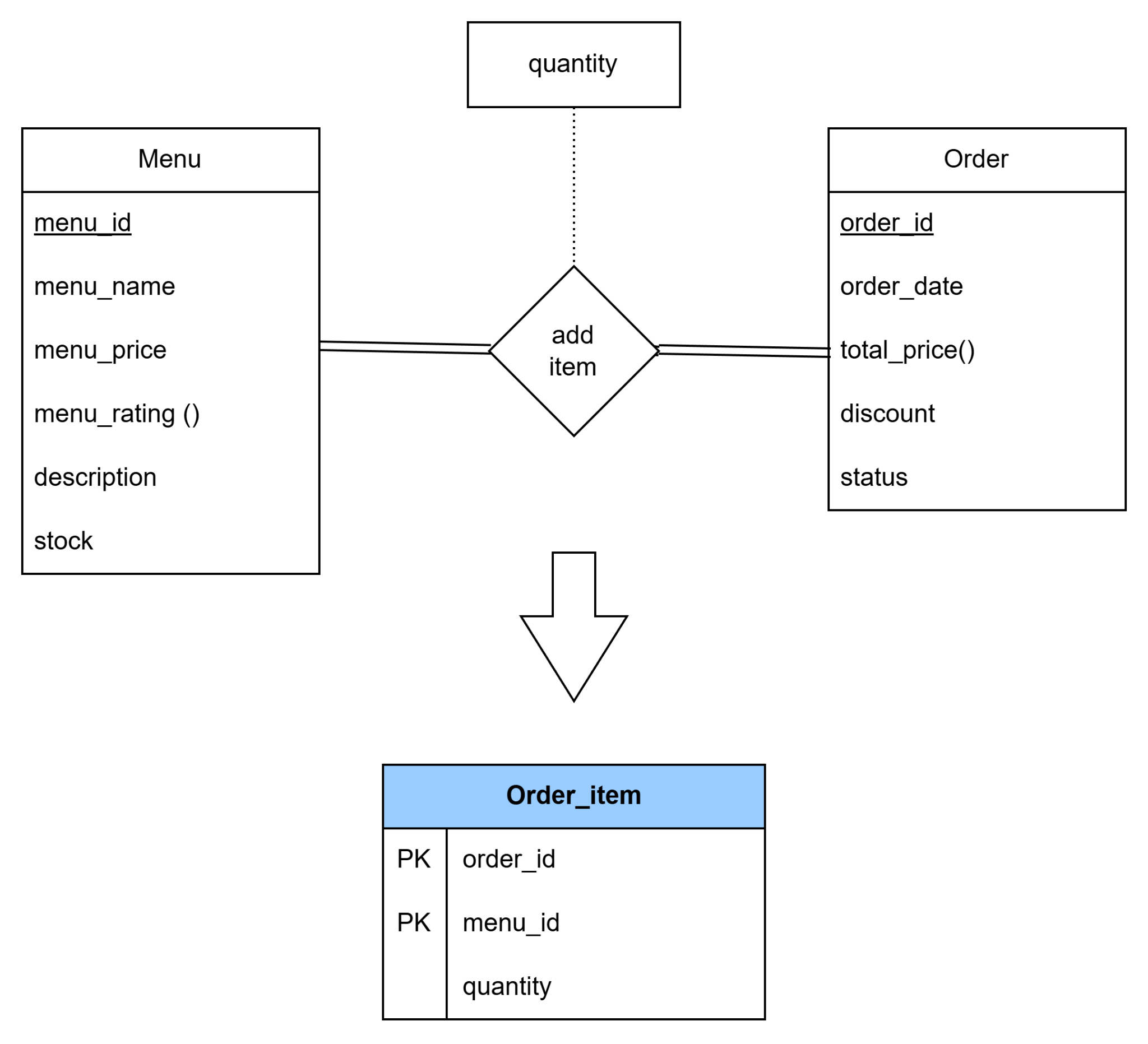
Pada *one-to-one relationship*, tidak perlu membuat relasi baru. Salah satu *primary key* ditambahkan pada salah satu relasi sebagai *foreign key*.

1. **One-to-many relationship**

****

Pada *one-to-many relationship*, tidak perlu membuat relasi baru. *Primary key* pada relasi “*one*” ditambahkan pada relasi “*many*” sebagai *foreign key*.

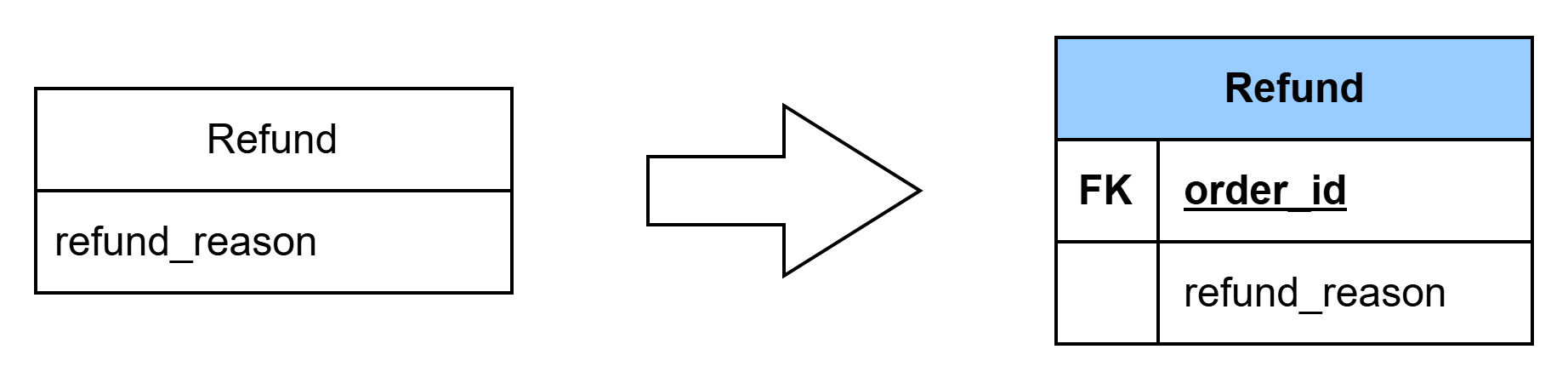
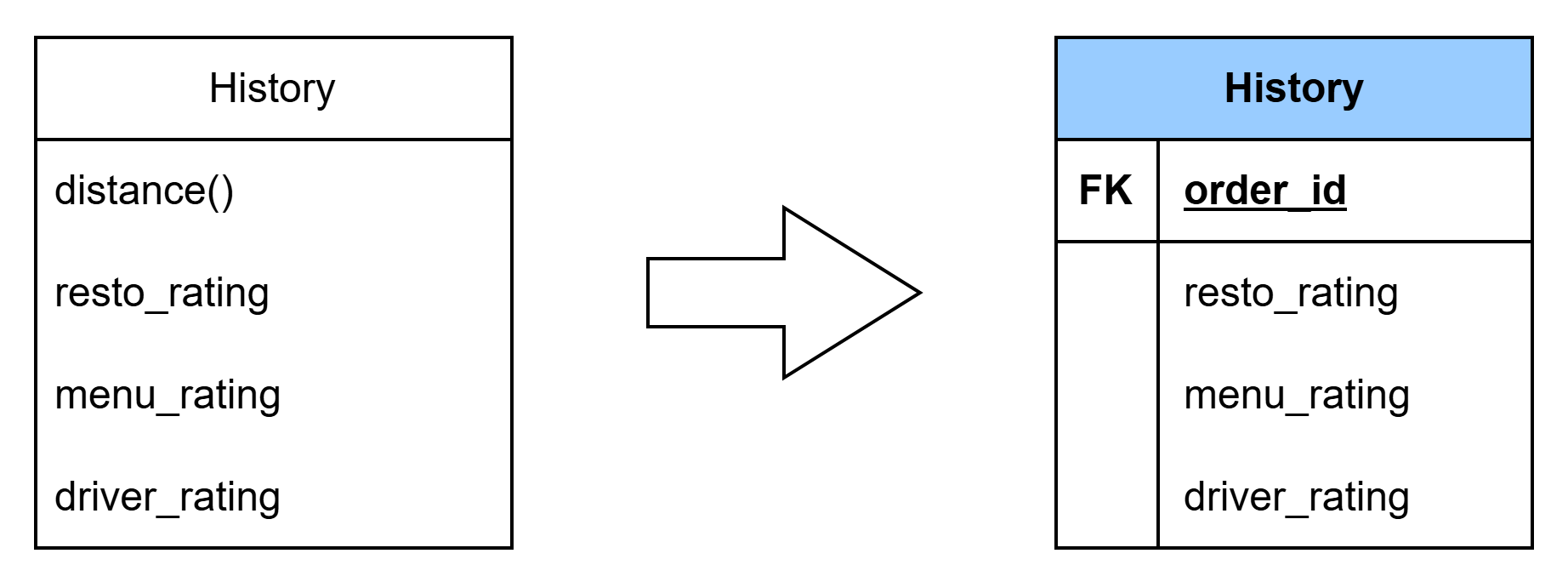
1. **Many-to-many relationship**

****

Pada *many-to-many relationship*, buat relasi baru untuk menggabungkan antara *primary key* kedua entity.

| *category = (category\_id, restaurant\_id)*  *order\_item = (menu\_id, order\_id, quantity)* |
| --- |

## **Pemetaan *specialization/generalization* menjadi relasi**

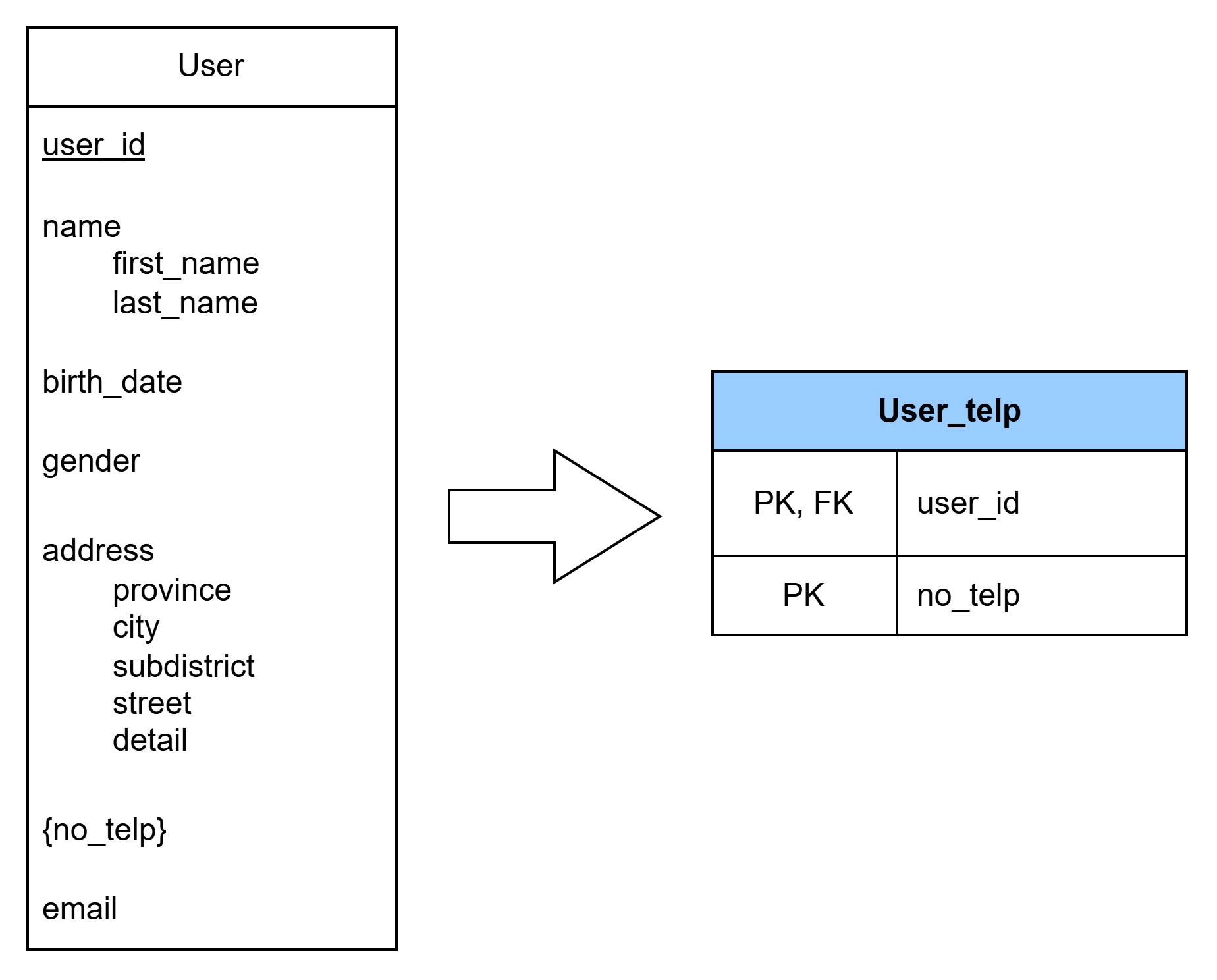
**

*Specialization* pada ERD Shopeepay kami yakni entity *Pesanan* menjadi *History* dan *Refund*,diubah menjadi relasi biasa pada ER-Model.

**Hasil relasi baru dari pemetaan *specialization/generalization* :**

| *history = (order\_id, resto\_rating, menu\_rating, driver\_rating)*  *refund = (order\_id, refund\_reason)* |
| --- |

## **Pemetaan *Multivalued Attributes***



Pada ER Model, *Multivalued attributes* perlu dibuat relasi baru untuk menghindari adanya duplikasi. Terdapat 1 *multivalued attributes* pada ER diagram kami, yaitu *no\_telp* pada entitas user. *Primary key* pada relasi baru tersebut terdiri dari *primary key parent* dan dirinya sendiri.

**Hasil relasi baru dari pemetaan *multivalued attributes* :**

| *user\_telp = (user\_id, no\_telp)* |
| --- |

## ***Derived Attributes***

*Derived Attributes* tidak dituliskan karena atribut ini tidak diimplementasikan dalam skema relasional. *Derived attributes* disajikan menggunakan *view.* Beberapa *Derived attributes* pada ER diagram kami yaitu sebagai berikut:

* 1. *Pesanan (total\_price())*
  2. *Restaurant (resto\_rating())*
  3. *Menu (menu\_rating())*
  4. *Driver (driver\_rating())*
  5. *History (distance())*

## ***Foreign Keys***

*Foreign keys* didefinisikan sebagai atribut dari suatu *entity* yang diambil dari *primary key entity* lain. *Foreign key* dapat terbentuk karena:

1. Penyisipan atribut akibat *one-to-one relationship*

| *Shopeepay (user\_id) → User(user\_id)*  *Transaction(order\_id) → Pesanan(order\_id)* |
| --- |

1. Penyisipan atribut akibat *one-to-many relationship* atau *many-to-one relationship*

| *Pesanan(driver\_id) → Driver (driver\_id)*  *Menu (resto\_id) → Restaurant (resto\_id)*  *Order\_address (order\_id) → Pesanan(order\_id)* |
| --- |

1. Penyisipan atribut akibat *multivalued attributes*

| *User\_telp (user\_id) → user (user\_id)* |
| --- |

## 

## **Hasil Pemetaan**

*Entity*

1. *user = (user\_id, first\_name, last\_name, birth\_date, gender, province, city, subdistrict, street, detail, email)*
2. *user\_telp = (user\_id, no\_telp)*
3. *shopeepay = (finance\_id, bank\_account, balance, user\_id)*
4. *transaction = (transaction\_id, transaction\_date, finance\_id, order\_id)*
5. *category = (category\_name, resto\_id, user\_id)*
6. *restaurant = (resto\_id, resto\_name, province, city, subdistrict, street)*
7. *menu = (menu\_id, menu\_name, menu\_price, description, stock, resto\_id)*
8. *pesanan = (order\_id, discount, status, driver\_id)*
9. *order\_item = (order\_id, menu\_id, quantity)*
10. *order\_address = (order\_address\_id, province, city, subdistrict, street, detail, order\_id)*
11. *driver = (driver\_id, driver\_name, no\_telp, vehicle\_number)*
12. *history = (order\_id, resto\_rating, menu\_rating, driver\_rating)*
13. *refund = (order\_id, refund\_reason)*

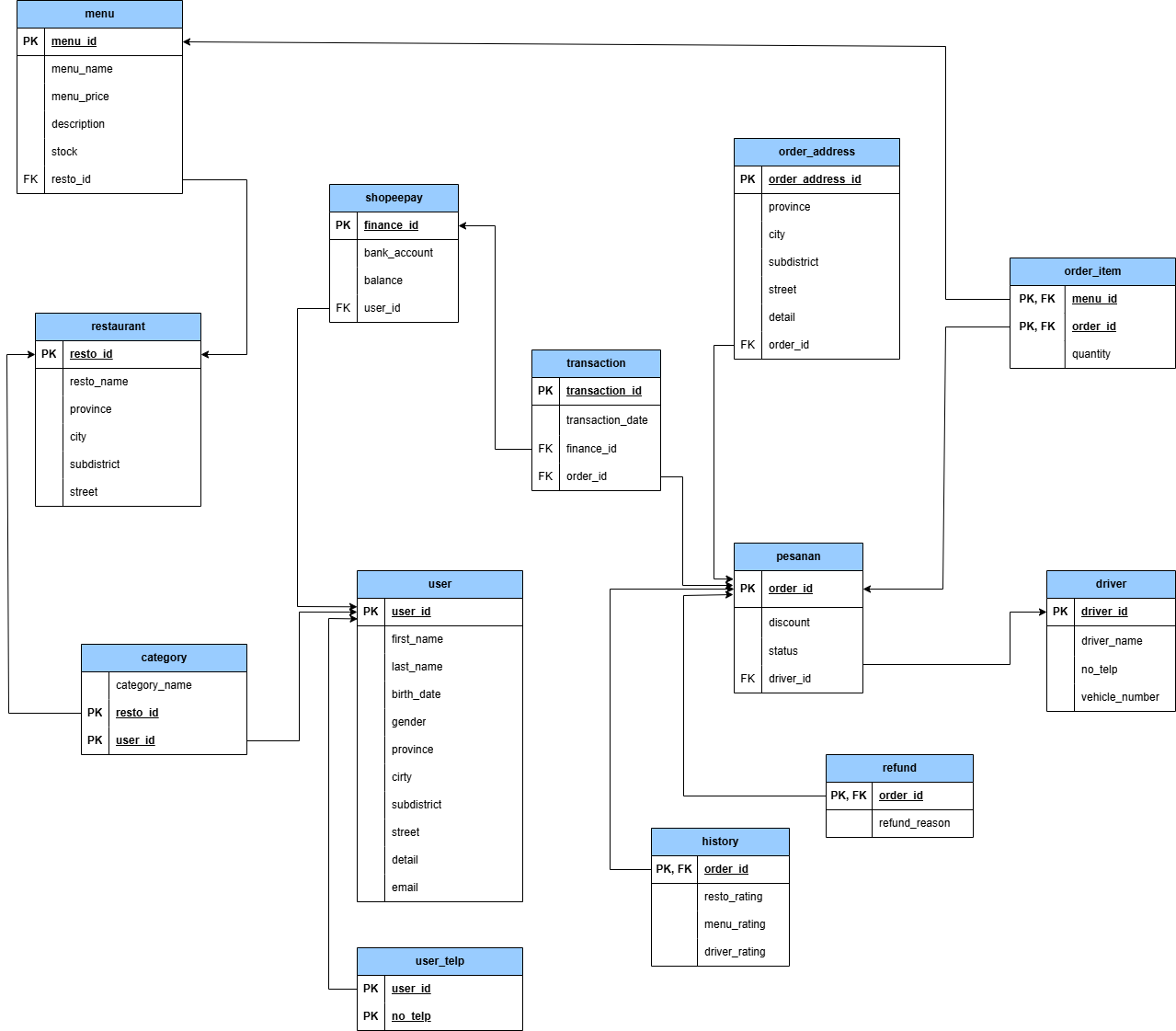
*Foreign Keys*

1. *shopeepay (user\_id) → user(user\_id)*
2. *transaction(order\_id) → pesanan(order\_id)*
3. *pesanan(driver\_id) → driver (driver\_id)*
4. *menu (resto\_id) → restaurant (resto\_id)*
5. *order\_address (order\_id) → pesanan(order\_id)*
6. *user\_telp (user\_id) → user (user\_id)*

# 

# 

# Diagram Model Relasional



*Gambar 3. Diagram Model Relasional*

# 

# Daftar Relasi

**Daftar Relasi dan Contoh Tuple**

## 5.1. Relasi User

*user = (user\_id, first\_name, last\_name, birth\_date, gender, province, city, subdistrict, street, detail, email)*

FK: -

Deskripsi:

1. *user\_id* adalah ID yang dimiliki setiap pengguna
2. *first\_name* adalah nama pertama dari setiap pengguna
3. *last\_name* adalah nama terakhir dari setiap pengguna
4. *birth\_date* adalah tanggal lahir pengguna. diformat dalam YYYY-MM-DD.
5. *gender* adalah jenis kelamin pengguna
6. *province, city, subdistrict, street,* dan *detail* adalah atribut komposit dari alamat pengguna
7. *email* adalah email pengguna

Keterangan:

* *gender* dibedakan menjadi 2 yaitu *Male* (laki-laki) dan *Female* (perempuan)
* *detail* terbagi menjadi 4 yaitu Apartment, Office, Home, Studio

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *user\_id* | varchar(3) | NO | PRIMARY |
| *first\_name* | varchar(15) | NO |  |
| *last\_name* | varchar(15) | NO |  |
| *birth\_date* | varchar(10) | NO |  |
| *gender* | varchar(6) | NO |  |
| *province* | varchar(15) | NO |  |
| *city* | varchar(15) | NO |  |
| *subdistrict* | varchar(15) | NO |  |
| *street* | varchar(50) | NO |  |
| *detail* | varchar(10) | NO |  |
| *email* | varchar(30) | NO |  |

| ***user\_***  ***id*** | ***first\_***  ***name*** | ***last\_***  ***name*** | ***birth\_***  ***date*** | ***gender*** | ***province*** | ***city*** | ***subdistrict*** | ***street*** | ***detail*** | ***email*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Nasim | Mandasari | 1982-10-07 | Male | Jawa Barat | Bandung | Cicalengka | Jalan Medokan Ayu No. 648 | Apartment | Nasim63@gmail.com |
| 2 | Kiandra | Pudjiastuti | 1976-06-04 | Male | DKI Jakarta | Jakarta Selatan | Pasar Minggu, | Jl. Jend. Sudirman No. 600 | Office | Kiandra35@gmail.com |
| 3 | Julia | Mustofa | 1989-08-15 | Female | Jawa Barat | Bandung | Bojongsoang | Gang Bangka Raya No. 9 | Office | Julia24@gmail.com |
| 4 | Eka | Tampubolon | 1979-03-04 | Male | Jawa Barat | Bandung | Bojongsoang | Gg. Tebet Barat Dalam No. 5 | Office | Eka33@gmail.com |
| 5 | Gamanto | Kuswandari | 1987-10-30 | Male | Jawa Barat | Bandung | Cicalengka | Gg. Sukabumi No. 5 | Office | Gamanto91@gmail.com |

## 5.2. Relasi User\_telp

*user\_telp = (user\_id, no\_telp)*

FK:

* *user\_telp (user\_id) → user (user\_id)*
* *user\_telp (no\_telp) → user (no\_telp)*

Deskripsi:

1. *user\_id* adalah ID unik dari setiap pengguna
2. *no\_telp* adalah nomor telepon pengguna

Keterangan:

* 1 user bisa memiliki lebih dari satu nomor telepon

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *user\_id* | varchar(3) | NO | PRIMARY |
| *no\_telp* | varchar(12) | NO | PRIMARY |

| ***user\_id*** | ***no\_telp*** |
| --- | --- |
| 1 | 087599277659 |
| 2 | 082167261594 |
| 3 | 086145832352 |
| 3 | 086860289249 |
| 4 | 087072214978 |

## 5.3. Relasi Shopeepay

*shopeepay = (finance\_id, bank\_account, balance, user\_id)*

FK: *shopeepay (user\_id) → User (user\_id)*

Deskripsi :

1. *finance\_id* adalah ID unik untuk setiap entitas keuangan.
2. *bank\_account* adalah nomor akun bank dalam format placeholder {}.
3. *balance* adalah saldo dalam satuan mata uang Rupiah.
4. *user\_id* adalah ID unik untuk setiap pengguna.

Keterangan : -

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *finance\_id* | varchar(4) | NO | PRIMARY |
| *bank\_account* | varchar(8) | NO |  |
| *balance* | int | NO |  |
| *user\_id* | varchar(3) | NO | FOREIGN |

| ***finance\_id*** | ***bank\_account*** | ***balance*** | ***user\_id*** |
| --- | --- | --- | --- |
| 5887 | 10314239 | 4923400 | 84 |
| 8092 | 10170363 | 435500 | 82 |
| 8796 | 43996200 | 1675400 | 42 |
| 7661 | 83852220 | 1903500 | 115 |
| 3674 | 73352585 | 4269900 | 51 |

## 5.4. Relasi Transaction

*Transaction = (transaction\_id, transaction\_date, finance\_id, user\_id)*

FK:

- *Transaction (finance\_id) → Shopeepay (finance\_id)*

*- Transaction (user\_id) → User (user\_id)*

Deskripsi :

1. *transaction\_id* adalah ID unik untuk setiap transaksi, dengan format 3 angka.
2. *transaction\_date* adalah tanggal transaksi, diformat dalam YYYY-MM-DD.
3. *finance\_id* adalah ID unik untuk setiap entitas keuangan.
4. *order\_id* adalah ID unik untuk setiap pemesanan.

Keterangan : -

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *transaction\_id* | varchar(4) | NO | PRIMARY |
| *transaction\_date* | varchar(10) | NO |  |
| *finance\_id* | varchar(4) | NO | FOREIGN |
| *order\_id* | vachar(11) | NO | FOREIGN |

| ***transaction\_id*** | ***transaction\_date*** | ***finance\_id*** | ***order\_id*** |
| --- | --- | --- | --- |
| W656 | 2022-5-21 | 7661 | DM-957-F7-U |
| N060 | 2021-2-12 | 1747 | AB-102-Z4-U |
| D774 | 2021-5-13 | 7185 | AZ-767-V7-U |
| K826 | 2022-6-12 | 5341 | GL-988-D0-U |
| H937 | 2024-2-18 | 3182 | SZ-782-X4-U |

## 

## 

## 

## 5.5. Relasi Category

category = *(category\_name, resto\_id, user\_id)*

FK:

- *category (resto\_id) → Restaurant (resto\_id)*

*- category (user\_id) → User (user\_id)*

Deskripsi:

* *category\_name* adalah nama kategori untuk restoran.
* *resto\_id* adalah ID unik milik restoran.
* *user\_id* adalah ID unik milik user.
* *resto\_id* dan *user\_id* adalah atribut komposit sebagai candidate key dari relasi *category*.

Keterangan:

* Hanya ada 9 kategori.
* Kategori dikelompokkan berdasarkan **menu** yang dijual di suatu restoran, bukan diskon dan lain-lain.

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *category\_name* | varchar(20) | NO |  |
| *resto\_id* | varchar(7) | NO | PRIMARY |
| *user\_id* | varchar(3) | NO | PRIMARY |

| ***category\_name*** | ***resto\_id*** | ***user\_id*** |
| --- | --- | --- |
| bakery | RES-141 | 18 |
| asian food | RES-099 | 125 |
| western food | RES-068 | 6 |
| fast food | RES-031 | 42 |
| noodles | RES-072 | 126 |

## 5.6. Relasi Restaurant

restaurant = *(resto\_id, resto\_name, province, city, subdistrict, street)*

FK: *-*

Deskripsi:

* *resto\_id* adalah ID unik untuk setiap restaurant.
* *resto\_name* adalah nama restoran.
* *province, city, subdistrict, street,* dan *detail* adalah atribut komposit dari alamat restoran

Keterangan:

* *address* berisi sub komponen: *province, city, subdistrict,* dan *street* spesifik di DKI Jakarta dan Jawa Barat: Bandung dan Sumedang.
* Restoran yang disediakan dalam database berjumlah 18 yang mewakili setiap kategori.

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *resto\_id* | varchar(7) | NO | PRIMARY |
| *resto\_name* | varchar(40) | NO |  |
| *province* | varchar(15) | NO |  |
| *city* | varchar(15) | NO |  |
| *subdistrict* | varchar(15) | NO |  |
| *street* | varchar(50) | NO |  |

| ***resto\_id*** | ***resto\_name*** | ***province*** | ***city*** | ***subdistrict*** | ***street*** |
| --- | --- | --- | --- | --- | --- |
| RES-001 | Crisbar | Jawa Barat | Bandung | Padalarang | Gg. Suryakencana No. 659 |
| RES-002 | KFC | Jawa Barat | Bandung | Sukajadi | Jl. Raya Setiabudhi No. 160 |
| RES-003 | MCDonald’s | Jawa Barat | Bandung | Antapani | Gang Medokan Ayu No. 744 |
| RES-004 | Ayam Geprek Bensu | Jawa Barat | Sumedang | Tanjungsari | Jalan Kiaracondong No. 802 |
| RES-005 | HokBen | DKI Jakarta | Jakarta Selatan | Pasar Minggu | Jl. Soekarno Hatta No. 027 |

## 

## 5.7. Relasi Menu

Menu = *(menu\_id, menu\_name, menu\_price, description, stock, resto\_id)*

FK: m*enu (resto\_id) → Restaurant (resto\_id)*

Deskripsi:

* *menu\_id* adalah ID unik untuk menu.
* *menu\_name* adalah nama menu.
* *menu\_price* adalah harga menu.
* *description* adalah deskripsi menu.
* *stock* adalah jumlah ketersediaan menu.

Keterangan:

* ID untuk setiap menu dijamin unik, karena walaupun menu yang dijual antar restoran bisa jadi sama, tetapi tetap dianggap sebagai speciality restoran masing-masing sehingga ID-nya berbeda.

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *menu\_id* | varchar(6) | NO | PRIMARY |
| *menu\_name* | varchar(25) | NO |  |
| *menu\_price* | int | NO |  |
| *description* | varchar(255) | NO |  |
| *stock* | int | NO |  |
| *resto\_id* | varchar(7) | NO | FOREIGN |

| ***menu\_id*** | ***menu\_name*** | ***menu\_price*** | ***description*** | ***stock*** | ***resto\_id*** |
| --- | --- | --- | --- | --- | --- |
| ME-001 | Bakso Kuah | 35000 | Makanan lezat dengan cita rasa autentik. | 17 | RES-630 |
| ME-002 | Ayam Geprek | 27000 | Menu yang pas untuk segala suasana. | 74 | RES-129 |
| ME-003 | Sop Iga | 99000 | Menu yang pas untuk segala suasana. | 45 | RES-137 |
| ME-004 | Kwetiau Siram | 52000 | Hidangan enak dan bergizi, siap dinikmati. | 44 | RES-107 |
| ME-005 | Gulai Ikan Patin | 121000 | Makanan lezat dengan cita rasa autentik. | 16 | RES-046 |

## 

## 

## 

## 

## 5.8. Relasi Pesanan

*pesanan = (order\_id, discount, status, driver\_id)*

FK: *Pesanan(driver\_id) → Driver(driver\_id)*

Deskripsi:

* *order\_id* adalah ID unik untuk setiap pemesanan.
* *discount* adalah jumlah/amount discount, dalam rentang 0-70 (persen).
* *status* adalah status pemesanan, “*done*” atau “*canceled*”.
* *driver\_id* adalah ID unik untuk driver.

Keterangan:

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *order\_id* | varchar(11) | NO | PRIMARY |
| *discount* | int | NO |  |
| *status* | varchar(8) | NO |  |
| *driver\_id* | varchar(6) | NO | FOREIGN |

| ***order\_id*** | ***discount*** | ***status*** | ***driver\_id*** |
| --- | --- | --- | --- |
| PU-204-Q3-U | 40 | Done | 294-15 |
| QX-011-L4-U | 20 | Done | 051-71 |
| GW-776-A5-U | 70 | Canceled | 471-55 |
| BA-497-K8-U | 70 | Done | 899-79 |
| YI-079-R7-U | 30 | Done | 614-00 |

## 5.9. Relasi Order Item

*Order\_item = (menu\_id, order\_id, quantity)*

FK:

* *Order\_item (menu\_id) → Menu (menu\_id)*
* *Order\_item (order\_id) → Pesanan (order\_id)*

Deskripsi:

* *menu\_id* adalah ID unik untuk setiap menu makanan/minuman.
* *order\_id* adalah ID unik untuk setiap pemesanan.
* *quantity* adalah jumlah makanan yang dipesan.

Keterangan:

* Minimal *quantity* ialah 1

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *menu\_id* | varchar(6) | NO | PRIMARY |
| *order\_id* | varchar(11) | NO | PRIMARY |
| *quantity* | int | NO |  |

| ***menu\_id*** | ***order\_id*** | ***quantity*** |
| --- | --- | --- |
| ME-019 | QX-606-E6-U | 1 |
| ME-39 | XI-634-N4-U | 8 |
| ME-119 | ZT-803-H3-U | 9 |
| ME-124 | FL-346-M0-U | 10 |
| ME-135 | WH-562-S5-U | 1 |

## 5.10. Relasi Order Address

*order\_address = (order\_address\_id, province, city, subdistrict, street, detail,*

*order\_id)*

FK: *order\_address (order\_id) → Pesanan (order\_id)*

Deskripsi:

* *province, city, subdistrict, street,* dan *detail* adalah atribut-atribut untuk setiap alamat pemesanan.
* *order\_id* adalah ID unik untuk setiap pemesanan.

Keterangan:

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *order\_address\_id* | varchar(3) | NO | PRIMARY |
| *province* | varchar(15) | NO |  |
| *city* | varchar(15) | NO |  |
| *subdistrict* | varchar(15) | NO |  |
| *street* | varchar(50) | NO |  |
| *detail* | varchar(9) | NO |  |
| *order\_id* | varchar(11) | NO | FOREIGN |

| ***Order\_***  ***address\_id*** | ***province*** | ***city*** | ***subdistrict*** | ***street*** | ***detail*** | ***order\_id*** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | DKI Jakarta | Jakarta Timur | Cipayung | Jl. Erlangga No. 923 | Apartment | CG-957-Y9-U |
| 2 | Jawa Barat | Bandung | Cicalengka | Gg. Dipenogoro No. 50 | Office | GP-377-N6-U |
| 3 | Jawa Barat | Bandung | Cicalengka | Gang Peta No. 1 | Office | YH-524-A6-U |
| 4 | DKI Jakarta | Jakarta Selatan | Cilandak | Gang Indragiri No. 485 | Apartment | MT-166-B5-U |
| 5 | DKI Jakarta | Jakarta Timur | Cipayung | Jl. Pacuan Kuda No. 8 | Home | IA-114-A5-U |

## 

## 5.11. Relasi Driver

driver =  *(driver\_id, driver\_name, no\_telp, vehicle\_number)*

FK: *-*

Deskripsi:

* *driver\_id* adalah ID unik untuk driver.
* *driver\_name* adalah nama driver.
* *no\_telp* adalah no. telp driver.
* *vehicle\_number* adalah nomor kendaraan yang unik untuk setiap driver.

Keterangan: -

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *driver\_id* | varchar(6) | NO | PRIMARY |
| *driver\_name* | varchar(20) | NO |  |
| *no\_telp* | varchar(12) | NO |  |
| *vehicle\_number* | varchar(10) | NO |  |

| ***driver\_id*** | ***driver\_name*** | ***no\_telp*** | ***vehicle\_number*** |
| --- | --- | --- | --- |
| 981-73 | Diah Prasetyo | 85471088680 | B 0355 HAM |
| 254-81 | Wisnu Samosir | 84479686625 | B 6583 LQA |
| 176-67 | Karna Putra | 87070849729 | B 7889 HWY |
| 386-33 | Setya Haryanto | 84311721848 | B 0710 HKB |
| 795-36 | Cager Hastuti | 89153722378 | B 3455 ZRD |

## 

## 

## 5.12. Relasi History

*history = (order\_id, resto\_rating, menu\_rating, driver\_rating)*

FK: *history(order\_id) → Pesanan(order\_id)*

Deskripsi:

* *order\_id* adalah id unik untuk tiap Pesanan.
* *resto\_rating* adalah rating untuk resto pada setiap pemesanan.
* *menu\_rating* adalah rating untuk suatu menu pada setiap pemesanan.
* *driver\_rating* adalah rating untuk driver pada setiap pemesanan.

Keterangan:

* Keseluruhan tuple berjumlah 271 sesuai dengan jumlah pesanan dengan status “Done”

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *order\_id* | varchar(11) | NO | PRIMARY |
| *resto\_rating* | int | NO |  |
| *menu\_rating* | int | NO |  |
| *driver\_rating* | int | NO |  |

| **order\_id** | **resto\_rating** | **menu\_rating** | **driver\_rating** |
| --- | --- | --- | --- |
| YK-831-I9-U | 3 | 5 | 3 |
| YT-773-U2-U | 3 | 2 | 5 |
| CW-038-Q7-U | 4 | 4 | 4 |
| AI-084-G7-U | 5 | 5 | 4 |
| GR-257-K7-U | 4 | 4 | 3 |

## 5.13. Relasi Refund

*refund= (order\_id, refund\_reason)*

FK: *history(order\_id) → Pesanan(order\_id)*

Deskripsi:

* *order\_id* adalah id unik untuk tiap Pesanan.
* *refund\_reason* adalah alasan pembatalan pesanan.

Keterangan:

* Keseluruhan tuple berjumlah 29 sesuai dengan jumlah pesanan dengan status “Cancelled”

| **Field** | **Type** | **Null** | **Key** |
| --- | --- | --- | --- |
| *order\_id* | varchar(11) | NO | PRIMARY |
| *refund\_reason* | varchar(50) | NO |  |

| ***order\_id*** | ***refund\_reason*** |
| --- | --- |
| YK-525-C6-U | Makanan tidak dapat dikonsumsi |
| VW-129-M8-U | Pesanan diterima dengan segel rusak |
| XV-066-G0-U | Pesanan diterima dengan segel rusak |
| AD-474-V7-U | Menerima makanan/minuman yang salah |
| IR-299-J2-U | Makanan tidak dapat dikonsumsi |

# 

# Daftar Pustaka

GeeksforGeeks. (n.d.). *Mapping from ER Model to Relational Model*. Diakses pada 26 November 2024, dari<https://www.geeksforgeeks.org/mapping-from-er-model-to-relational-model/>.

Tim Pengajar IF2040 Semester I TA 2024/2025. (2024). *Reducing E-R Diagrams to Relational Schemas.* Institut Teknologi Bandung.

# 

# Pembagian Tugas

| **Nama Lengkap - NIM** | **Deskripsi Tugas** |
| --- | --- |
| Anggita Najmi Layali - 18223122 | * Menghubungi asisten dan menjadwalkan asistensi * Mengisi dokumen bagian pemetaan *relationship* menjadi relasi, *foreign keys* * Membuat Model Relasional * Membuat revisi *milestone* 1 * Mengisi dokumen bagian daftar relasi *Restaurant, Menu, Category, Pesanan* * Membuat dummy data *Restaurant, Menu,* dan *Category* |
| Muhammad Dzaky Atha F - 18223124 | * Mengisi dokumen bagian pemetaan *entity* menjadi relasi, *multivalued attributes*, *derived attributes* * Membuat Model Relasional * Mengisi dokumen bagian daftar relasi *User, User\_telp, Order\_item, Order\_address* * Membuat dummy data *User, Order\_item, Order\_address* * Membuat SQL *dump* |
| M Rabbani K A - 18223130 | * Membuat Model Relasional * Mengisi dokumen bagian daftar relasi *History, Refund* * Membuat dummy data *Pesanan, History, Refund* |
| Sonya Putri Fadilah - 18223138 | * Membuat revisi *milestone* 1 * Membuat Model Relasional * Mengisi dokumen bagian daftar relasi *Shopeepay, Transaction, Driver* * Membuat dummy data *Shopeepay, Transaction, Driver* |

# Lampiran

## Moments of Meeting

**Asistensi Milestone 2**

| **Hari** | Kamis | **Tanggal** | 28 November 2024 |
| --- | --- | --- | --- |
| **Kelas** | 03 | **Kelompok** | 05 |
| **Waktu** | 20.30 -21.05 | **Tempat** | Google Meet |
| **Anggota Kelompok** | **NIM** | **Nama** | |
| 18223122 | Anggita Najmi Layali | |
| 18223124 | Muhammad Dzaky Atha F | |
| 18223130 | M Rabbani K A | |
| 18223138 | Sonya Putri Fadilah | |
| **Nama Asisten** | Anindita Widya Santoso | | |

| **Hasil Asistensi** | | |
| --- | --- | --- |
| **No.** | **Pertanyaan** | **Jawaban** |
|  | Untuk atribut yang hilang seperti derived dan multivalued attributes apakah benar-benar hilang (tidak dicantumkan di relational model)? | Ya, seharusnya dapat dibuat view tapi tidak perlu sekompleks itu, cukup nanti bisa dihitung di SQL-nya saja. |
| 2. | SQL dummy apakah harus sesuai dengan realita? | Tidak harus se-real life mungkin. Bisa saja restoran A menjual makanan yang tidak sesuai dengan di dunia nyata, tetapi asal masih relevan (sama-sama makanan). |
| 3. | Di ERD, user terhubung ke restaurant untuk memilih menu, apakah di ER-Model juga? Tapi terlihat kurang nyambung. | Ya, harus. Relationship menjadi relasi baru yang memuat user\_id dan resto\_id. |

| **Dokumentasi** |
| --- |
|  |