IF2040 Pemodelan Basis Data  
*ER Modeling*

**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**](#_v53hx7qnz47f)

[**1. Deskripsi Sistem 3**](#_rauwu15ztcex)

[**2. ER Diagram 5**](#_94b5k2hpjnds)

[**Pembagian Tugas 8**](#_1xi8gcvdm63r)

[**Lampiran 9**](#_leqja79r9mr1)

# 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.

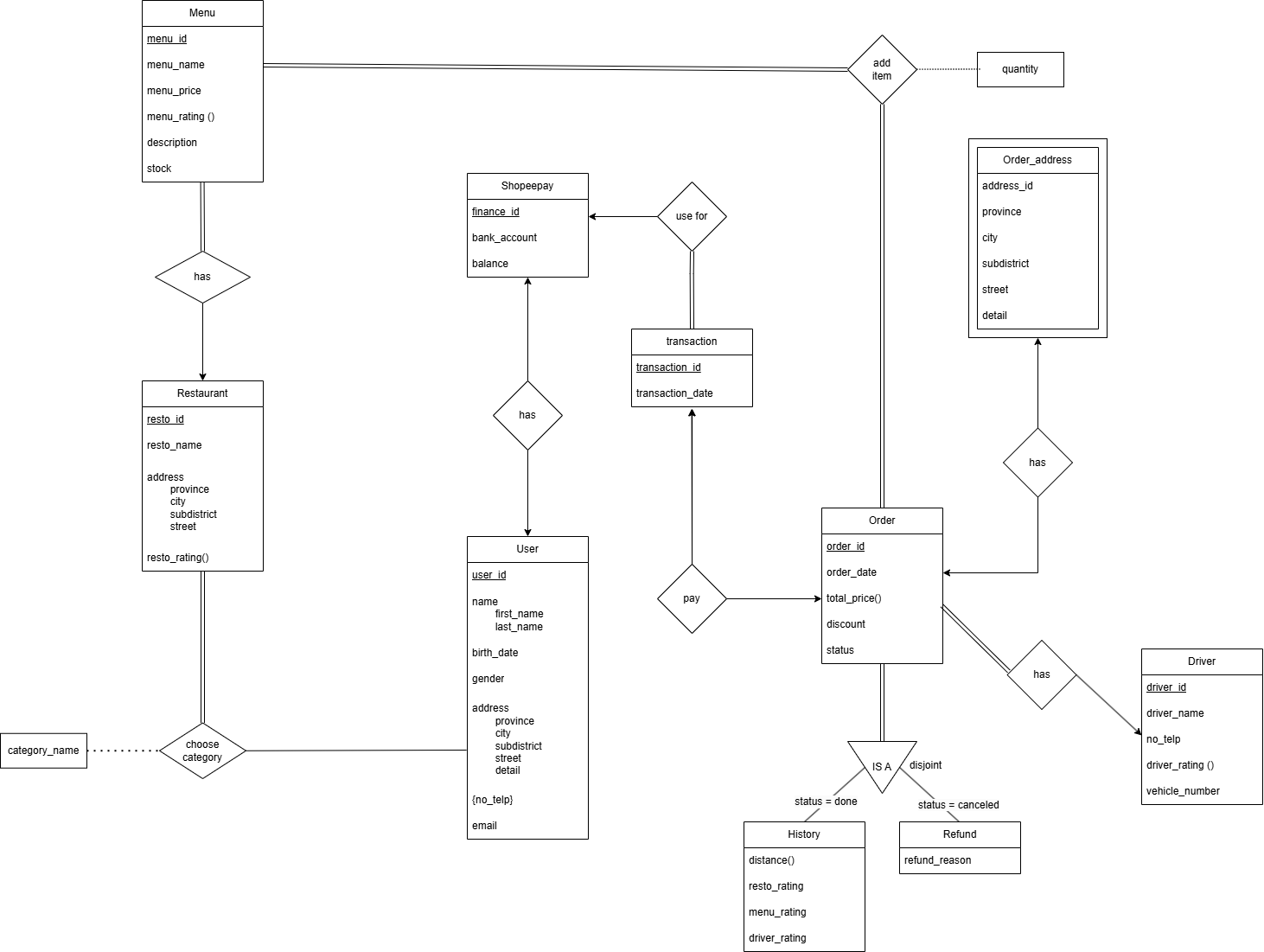
Setiap *user* dapat melakukan lebih dari satu pemesanan. Setiap pesanan akan dicatat ID pesanan yang unik, tanggal pemesanan, status penggunaan *voucher*, total harga yang dikalkulasikan dari harga tiap makanan yang dipesan, diskon, dan status pemesanan yang menyatakan apakah Order yang dilakukan selesai 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 juga 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 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

Link ER Diagram: [ERD\_Shopeefood.drawio - Google Drive](https://drive.google.com/file/d/1Wp14ncG1fd8VKXi-V8xh7r4_Gw1OOcc5/view)



Gambar 2. Diagram ER

Top(\*S) -=

| No. | Entity Set | Atribut |
| --- | --- | --- |
| 1. | User | user\_id, name (composite), birth\_date, gender, address(composite), no\_telp (multivalued variable), email |
| 2. | Menu | menu\_id, menu\_name, menu\_price, menu\_rating(), description, stock |
| 3. | Restaurant | resto\_id, resto\_name, address (composite), resto\_rating() |
| 4. | Shopeepay | finance\_id, bank\_account (multivalued variable), balance |
| 5. | History (specialization) | distance(), resto\_rating, menu\_rating, driver\_rating |
| 6. | Order | order\_id, order\_date, discount, total\_price(), status |
| 7. | Transaction | transaction\_id, transaction\_date (composite) |
| 8. | Order\_address (weak) | order\_addres\_id, province, city, subdistrict, street, detail |
| 9. | Driver | driver\_id, driver\_name, driver\_rating(), vehicle\_number, no\_telp |
| 10. | Refund (specialization) | refund\_reason |

Tabel 1. Daftar Entity Set dan Atributnya

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 order dan memilih 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 *order*, 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.

*Derived Attributes*

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

# 

# 

# Pembagian Tugas

| **Nama Lengkap - NIM** | **Deskripsi Tugas** |
| --- | --- |
| Anggita Najmi Layali - 18223122 | * Mengisi dokumen bagian deskripsi * Mencari entity set dan atributnya, serta jenis relationship antar entity * Mengisi bagian asumsi * Membuat ERD |
| Muhammad Dzaky Atha F - 18223124 | * Mengisi dokumen bagian deskripsi * Melakukan formatting laporan * Mencari entity set dan atributnya * Mengisi bagian asumsi * Membuat ERD |
| M Rabbani K A - 18223130 | * Membuat keterangan * Mengisi bagian asumsi * Membuat ERD |
| Sonya Putri Fadilah - 18223138 | * Mencari entity set dan atributnya, serta jenis relationship antar entity * Mengisi bagian asumsi * Membuat ERD |

# Lampiran

## Moments of Meeting

**Asistensi Milestone 1**

| **Hari** | Jum’at | **Tanggal** | 15 November 2024 |
| --- | --- | --- | --- |
| **Kelas** | 03 | **Kelompok** | 05 |
| **Waktu** | 14.00 - 14.15 | **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** |
| 1. | \*Koreksi-koreksi | * Entity kebanyakan. Simplified dan hapus entity yang tidak perlu. Jika terlalu banyak entity, implementasi di SQL-nya bakal menyulitkan * Tidak perlu memakai angka pada garis untuk menyatakan jenis relationship antar entity * Hapus entity yang tidak perlu seperti favourite, cart, dan history * Overall sudah bagus |
| 2. | Apakah sistemnya harus persis seperti aslinya? | Tidak perlu persis. Kalau sistemnya terlalu kompleks takutnya malah menyulitkan |
| 3. | Konsep ternary seperti apa? | Punya 1 relationship yang menyambung ke 3 entity |

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