IF2040 Pemodelan Basis Data  
Ketergantungan Fungsional dan Normalisasi

**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**](#_gbx1lzlmntaw)

[**1. Deskripsi Sistem 3**](#_flb7x01t0zue)

[**2. Diagram Model Relasional 5**](#_gqsdgzwumsew)

[**3. Daftar Functional Dependencies, Candidate Key, dan Bentuk Normal 6**](#_km5hai20ran2)

[1. Relasi User 6](#_pek9evbz827p)

[2. Relasi User\_telp 7](#_xxodvh36b9vj)

[3. Relasi Shopeepay 7](#_ysdu0grysg7w)

[4. Relasi Transaction 8](#_yhj4or6fdhsa)

[5. Relasi Category 8](#_76iyg2lqmaop)

[6. Relasi Restaurant 9](#_9gsugd116y6p)

[7. Relasi Menu 9](#_r8gcit4kq8rm)

[8. Relasi Pesanan 10](#_1dfoz4ahm0db)

[9. Relasi Order Item 11](#_juax24nqsuk0)

[10. Relasi Order Address 11](#_ra1tduch30ou)

[11. Relasi Driver 12](#_awj701wx6mlv)

[12. Relasi History 12](#_3o3rmcq2okc)

[13. Relasi Refund 13](#_dbpbjjjaef9u)

[**4. Hasil dari Normalisasi 14**](#_axvtucd5dm6v)

[4.1. Relasi Menu 14](#_iyeei9g8xt56)

[**4. Daftar Informasi dari Basis Data 16**](#_zi7bnendk0jm)

[**Daftar Pustaka 27**](#_9tinbd4u5g6t)

[**Pembagian Tugas 28**](#_ylsct3m7f17m)

[**Lampiran 29**](#_vk1krr8pagf6)

[Moments of Meeting 29](#_tc6o44porpe)

# 

# 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.

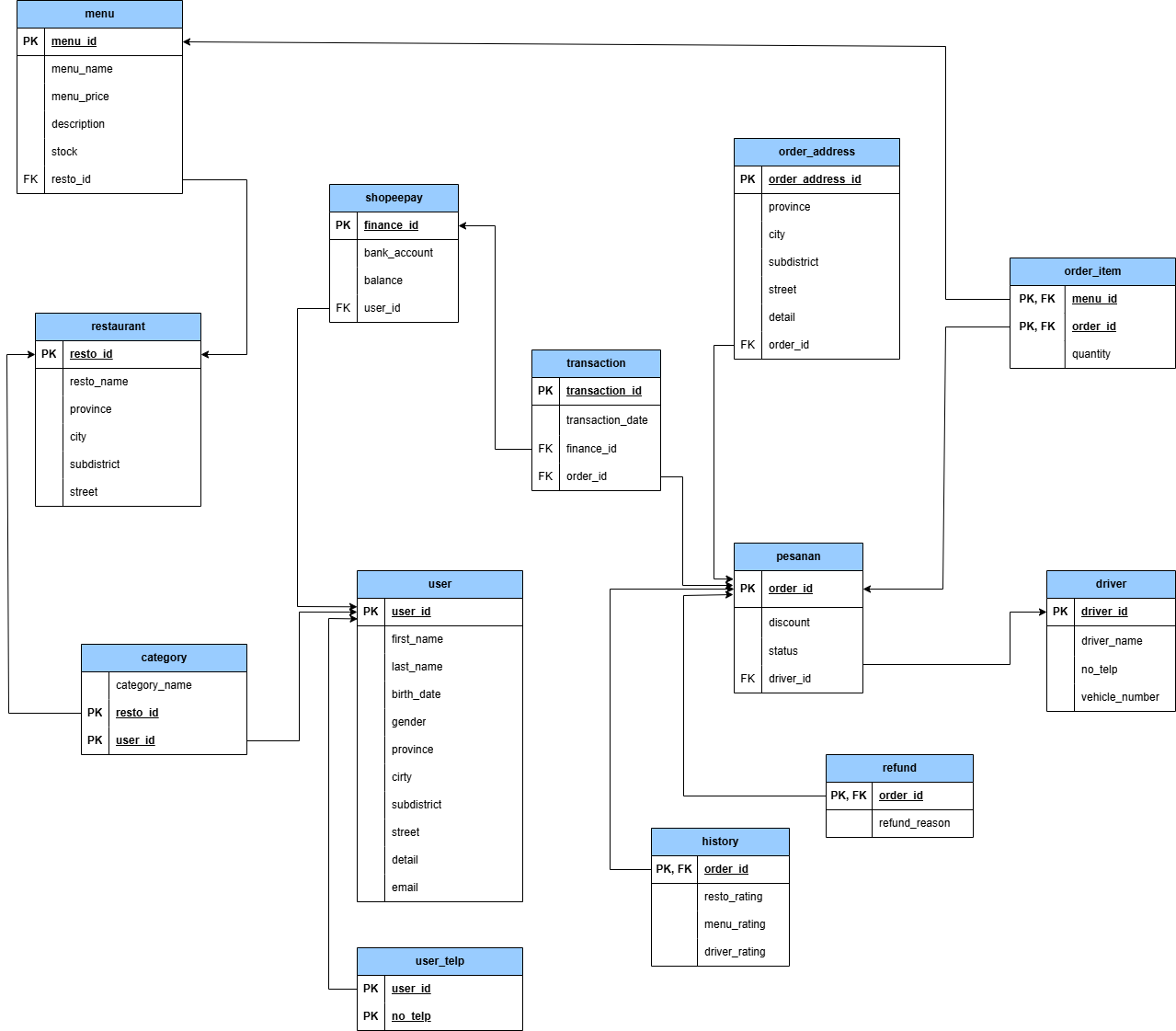
*User* juga dapat mencari menu makanan yang tersedia baik dengan langsung mencari nama makanan tersebut, maupun melihat-lihat melalui restoran. Setiap menu memiliki ID menu yang unik, nama menu, rating menu (1-5) yang dikalkulasi dari rata-rata rating menu oleh seluruh *user* tiap kali melakukan pemesanan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

# Diagram Model Relasional



Gambar 2. Diagram Model Relasional

# 

# Daftar *Functional Dependencies*, *Candidate Key*, dan Bentuk Normal

## Relasi User

FD = {*user\_id → first\_name, last\_name, birth\_date, gender, province, city,*

*subdistrict, street, detail, email*;

*email* →*user\_id,*  *first\_name, last\_name, birth\_date, gender, province,*

*city, subdistrict, street, detail}*

*Candidate Key* = {*user\_id*, *email*}

Relasi User sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *candidate key*, yaitu *user\_id* dan *email*. Pada atribut *first\_name, last\_name, birth\_date, gender province, city, subdistrict, street dan detail* bisa saja memiliki value yang sama sehingga tidak dapat menjadi *candidate key*.

| ***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 |

## Relasi User\_telp

FD = {*user\_id*, *no\_telp → user\_id, no\_telp*}

*Candidate Key* = {*user\_id*, *no\_telp*}

Relasi User\_telp sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *candidate key*, yaitu *user\_id* dan *no\_telp*.

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

## Relasi Shopeepay

FD = {*finance\_id*→ *bank\_account, balance, user\_id;*

*bank\_account* → *finance\_id, balance, user\_id;*

*user\_id → finance\_id, bank\_acount, balance*}

*Candidate Key* = {*finance\_id, bank\_account, user\_id*}

Relasi Shopeepay sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *candidate key*, yaitu *finance\_id, bank\_account,* dan *user\_id*. Pada atribut *balance* bisa saja terdapat value yang sama di sehingga tidak dapat menjadi *candidate key*

| ***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 |

## Relasi Transaction

FD = {*transaction\_id* → *transaction\_date, finance\_id, order\_id;*

*order\_id → transaction\_id, transaction\_date, finance\_id*}

*Candidate Key* = {*transaction\_id, order\_id*}

Relasi Transaction sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung kepada *candidate key*, yaitu *transaction\_id* dan *order\_id*. Pada atribut *finance\_id* bisa saja terdapat value yang sama sehingga tidak dapat menjadi *candidate key.*

| ***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 |

## 

## Relasi Category

FD = {*resto\_id, user\_id → category\_name*}

Candidate key = {*resto\_id, user\_id*}

Relasi Category sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *resto\_id* dan *user\_id*. Pada atribut *category\_name* bisa saja terdapat value yang sama di sehingga tidak dapat menjadi *candidate key.*

| **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 |

## Relasi Restaurant

FD = {*resto\_id → resto\_name, province, city, subdistrict, street*}

Candidate key = {*resto\_id*}

Relasi Restaurant sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *resto\_id*. Pada atribut *resto\_name, province, city, subdistrict,* dan *street* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| **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 |

## 

## Relasi Menu

FD = {*menu\_id → menu\_name, menu\_price, description, stock;*

*resto\_id → menu\_id*}

Candidate key = {*menu\_id*}*,* {*resto\_id*}

Relasi Menu berada pada bentuk 2NF karena atribut *menu\_name, menu\_price, description, stock* bergantung pada *candidate key* yakni *menu\_id*. Akan tetapi, atribut *menu\_id* bergantung pada *candidate key* lain yaitu *resto\_id* sehingga masih perlu normalisasi untuk mencapai bentuk normal yang lebih tinggi.

| **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 |

## 

## Relasi Pesanan

FD : {*order\_id → discount, status, driver\_id*}

Candidate key : {*order\_id*}

Relasi Pesanan sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *order\_id*. Pada atribut *discount, status,* dan *driver\_id* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| **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 |

## 

## Relasi Order Item

FD = {*menu\_id, order\_id → quantity*}

*Candidate Key* = { *menu\_id, order\_id*}

Relasi Order Item sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *menu\_id* dan *order\_id*. Pada atribut *quantity* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| ***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 |

## Relasi Order Address

FD = {*order\_address\_id, → province, city, subdistrict, street, detail;*

*order\_id → province, city, subdistrict, street, detail*}

*Candidate Key* = {*order\_address\_id, order\_id* }

Relasi Order Address berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key* yakni *order\_address\_id* dan *order\_id*. Pada atribut *province, city, subdistrict, street, dan detail* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| ***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 |

## 

## Relasi Driver

FD = {*driver\_id* → *driver\_name, no\_telp, vehicle\_number*}

*Candidate Key* = {*driver\_id*}

Relasi Order Item sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *menu\_id* dan *order\_id*. Pada atribut *quantity* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| ***driver\_id*** | ***driver\_name*** | ***no\_telp*** | ***vehicle\_number*** |
| --- | --- | --- | --- |
| 981-73 | Diah Prasetyo | 085471088680 | B 0355 HAM |
| 254-81 | Wisnu Samosir | 084479686625 | B 6583 LQA |
| 176-67 | Karna Putra | 087070849729 | B 7889 HWY |
| 386-33 | Setya Haryanto | 084311721848 | B 0710 HKB |
| 795-36 | Cager Hastuti | 089153722378 | B 3455 ZRD |

## 

## 

## Relasi History

FD : {*order\_id → resto\_rating, menu\_rating, driver\_rating*}

Candidate key : {*order\_id*}

Relasi History sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *order\_id*. Pada atribut *resto\_rating, menu\_rating,* dan *driver\_rating* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| ***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 |

## 

## Relasi Refund

FD : {*order\_id → refund\_reason*}

Candidate key : {*order\_id*}

Relasi Refund sudah berada pada bentuk BCNF karena semua *functional dependencies* pada relasi ini bergantung pada *candidate key*, yakni *order\_id*. Pada atribut *refund\_reason* bisa saja terdapat value yang sama sehinggatidak dapat menjadi *candidate key*.

| ***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 |

# 

# 4. Hasil dari Normalisasi

## 4.1. Relasi Menu

Untuk mengubah relasi menu menjadi bentuk normal yang lebih tinggi, maka dipecah menjadi relasi *Menu* dan relasi *Menu-Resto* dengan spesifikasi sebagai berikut.

1. **Relasi Menu**

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

## 

## 

## 

## 

## 

## 

## 

FD = {*menu\_id → menu\_name, menu\_price, description, stock*}

*Candidate Key* = {*menu\_id*}

Relasi *Menu* kini berada pada bentuk BCNF karena semua *functional dependencies* pada relasi *Menu* bergantung pada *superkey,* yaitu atribut *menu\_id*.

1. **Relasi Menu-Resto**

| ***menu\_id*** | ***resto\_id*** |
| --- | --- |
| ME-001 | RES-063 |
| ME-002 | RES-129 |
| ME-003 | RES-137 |
| ME-004 | RES-107 |
| ME-005 | RES-046 |

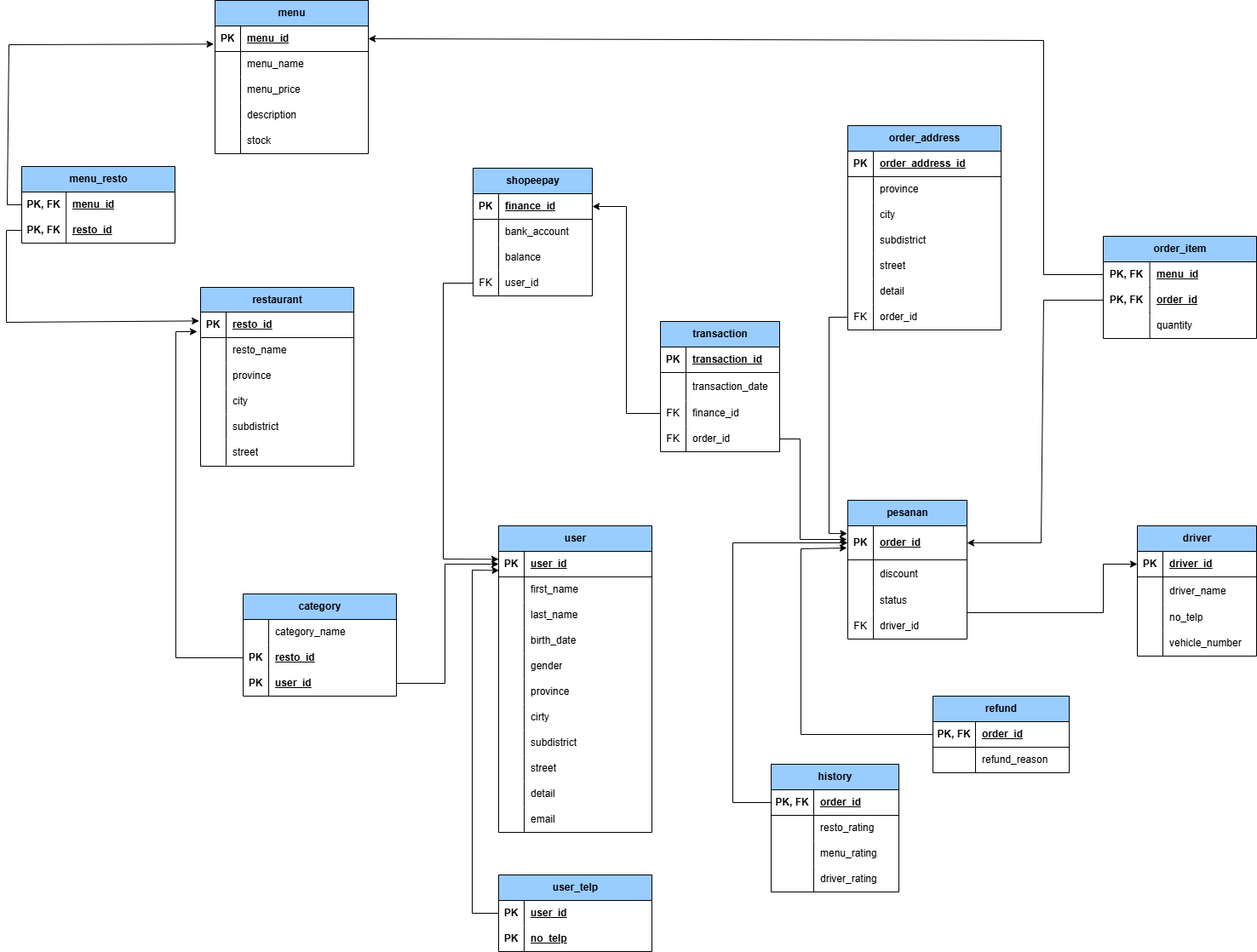
## 

FD = {*resto\_id → menu\_id*}

*Candidate Key* = {*resto\_id*}

Relasi *Menu-Resto* kini berada pada bentuk BCNF karena semua *functional dependencies* pada relasi *Menu-Resto* bergantung pada *superkey,* yaitu atribut *resto\_id*.

**Diagram Relasional Model Hasil Normalisasi**



# Daftar Informasi dari Basis Data

| **Informasi 1** | Daftar user yang memiliki lebih dari 1 nomor telepon |
| --- | --- |
| ***Query*** | SELECT concat(u.first\_name, " ", u.last\_name) as nama\_user, u.user\_id, count(ut.no\_telp) as jumlah\_no\_telp  FROM user u  RIGHT OUTER JOIN user\_telp ut ON u.user\_id = ut.user\_id  GROUP BY u.user\_id  HAVING jumlah\_no\_telp > 1; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | User, User\_telp |

| **Informasi 2** | Daftar nama 30 restoran dengan rating tertinggi |
| --- | --- |
| ***Query*** | SELECT r.resto\_name, ROUND(AVG(h.resto\_rating), 1) AS avg\_resto\_rating  FROM history h NATURAL JOIN pesanan p  NATURAL JOIN order\_item i NATURAL JOIN menu m  NATURAL JOIN menu\_resto mr NATURAL JOIN restaurant r  GROUP BY r.resto\_name  ORDER BY avg\_resto\_rating DESC  LIMIT 30; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | History, pesanan, order\_item, menu, menu\_resto, restaurant |

| **Informasi 3** | Daftar rating driver yang melayani lebih dari 2 order dari daerah di Jawa Barat. |
| --- | --- |
| ***Query*** | SELECT d.driver\_name, ROUND(AVG(h.driver\_rating), 1) AS avg\_driver\_rating, a.subdistrict  FROM driver d  NATURAL JOIN pesanan p  NATURAL JOIN history h  NATURAL JOIN order\_address a  WHERE d.driver\_id IN (  SELECT p.driver\_id  FROM pesanan p  GROUP BY p.driver\_id  HAVING COUNT(p.order\_id) > 2 )  AND a.province = 'Jawa Barat'  GROUP BY d.driver\_name, a.subdistrict  ORDER BY avg\_driver\_rating DESC |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | Driver, pesanan, history, order\_address |

| **Informasi 4** | Order\_id yang bertransaksi menggunakan shopeepay dan berstatus refund. |
| --- | --- |
| ***Query*** | SELECT order\_id, refund\_reason  FROM refund  WHERE order\_id IN (  SELECT order\_id FROM pesanan )  UNION  SELECT order\_id, refund\_reason  FROM refund; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | Refund, pesanan, transaction, shopeepay |

| **Informasi 5** | Total harga pesanan yang dipesan oleh user bernama belakang “Hidayanto”. |
| --- | --- |
| ***Query*** | SELECT  u.user\_id,  CONCAT(u.first\_name, ' ', u.last\_name) AS user\_name,  SUM(m.menu\_price \* oi.quantity) AS total\_harga\_pesanan  FROM User u  JOIN category c ON u.user\_id = c.user\_id  JOIN restaurant r ON c.resto\_id = r.resto\_id  JOIN menu\_resto mr ON r.resto\_id = mr.resto\_id  JOIN menu m ON mr.menu\_id = m.menu\_id  JOIN order\_item oi ON m.menu\_id = oi.menu\_id  JOIN pesanan p ON oi.order\_id = p.order\_id  WHERE u.last\_name = 'Hidayanto'  GROUP BY u.user\_id, u.first\_name, u.last\_name; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | User, category, restaurant, menu\_resto, menu, order\_item, pesanan |

# 

| **Informasi 6** | Daftar driver dengan rating kurang dari 3 |
| --- | --- |
| **Query** | SELECT d.driver\_name, AVG(h.driver\_rating) AS avg\_driver\_rating  FROM driver d  NATURAL JOIN pesanan p  NATURAL JOIN history h  GROUP BY d.driver\_name  HAVING avg\_driver\_rating < 3  ORDER BY avg\_driver\_rating |
| **Hasil Query** |  |
| **Tabel yang terlibat** | driver, pesanan, history |

# 

# 

| **Informasi 7** | Daftar jumlah user per-*subdistrict* |
| --- | --- |
| **Query** | SELECT u.subdistrict, count(u.subdistrict) as total\_user\_city  FROM user u  GROUP BY u.subdistrict |
| **Hasil Query** |  |
| **Tabel yang terlibat** | user |

| **Informasi 8** | Daftar 10 user dengan pesanan yang mendapat persentase diskon paling besar |
| --- | --- |
| **Query** | SELECT p.order\_id,  CONCAT(u.first\_name, ' ', u.last\_name) AS user\_name,  SUM(oi.quantity \* m.menu\_price) AS price\_before,  p.discount AS disc\_percentage,  ROUND(SUM(oi.quantity \* m.menu\_price) \* (p.discount / 100), 2) AS disc\_value,  ROUND(SUM(oi.quantity \* m.menu\_price) \* (1 - (p.discount / 100)), 2)  AS price\_after  FROM pesanan p  JOIN order\_item oi ON p.order\_id = oi.order\_id  JOIN menu m ON oi.menu\_id = m.menu\_id  JOIN transaction t ON p.order\_id = t.order\_id  JOIN shopeepay sp ON t.finance\_id = sp.finance\_id  JOIN user u ON sp.user\_id = u.user\_id  GROUP BY  p.order\_id, u.user\_id, u.first\_name, u.last\_name, p.discount  ORDER BY p.discount DESC  LIMIT 10; |
| **Hasil Query** |  |
| **Tabel yang terlibat** | pesanan, order\_item, menu, transaction, shopeepay, user |

| **Informasi 9** | Daftar 10 menu terlaris berdasarkan jumlah pemesanan dan total pendapatan penjualan |
| --- | --- |
| **Query** | SELECT  m.menu\_name,  SUM(oi.quantity) AS total\_quantity\_sold,  SUM(oi.quantity \* m.menu\_price) AS total\_revenue  FROM  menu m  JOIN  order\_item oi ON m.menu\_id = oi.menu\_id  GROUP BY  m.menu\_id, m.menu\_name  ORDER BY  total\_quantity\_sold DESC  LIMIT 10; |
| **Hasil Query** |  |
| **Tabel yang terlibat** | menu, order\_item |

| **Informasi 10** | Daftar 10 user yang memiliki saldo awal shopeepay terbesar |
| --- | --- |
| ***Query*** | SELECT concat(u.first\_name, " ", u.last\_name) as nama\_user,  sp.finance\_id, sp.balance as saldo\_sekarang,  SUM(quantity \* menu\_price) AS total\_harga\_keseluruhan\_pesanan\_user,  sp.balance + SUM(quantity \* menu\_price) AS saldo\_awal  FROM history h  NATURAL JOIN transaction t  NATURAL JOIN shopeepay sp  NATURAL JOIN order\_item  NATURAL JOIN menu  NATURAL JOIN user u  GROUP BY sp.finance\_id, sp.balance  ORDER BY saldo\_awal desc  limit 10; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | History, transaction, shopeepay, order\_item, menu, user |

**BONUS**

| **Informasi 11** | Daftar restoran di Bandung yang menjual menu “ayam” dan rating menunya. Apabila rating > 3, tampilkan sebagai “rating tinggi” sedangkan rating < 3 sebagai “rating rendah”. |
| --- | --- |
| ***Query*** | SELECT DISTINCT  r.resto\_name,  m.menu\_name,  CASE  WHEN i.menu\_id IN (  SELECT i.menu\_id  FROM order\_item i  NATURAL JOIN pesanan p  NATURAL JOIN history h  GROUP BY i.menu\_id  HAVING AVG(h.menu\_rating) > 3  ) THEN 'Rating Tinggi'  ELSE 'Rating Rendah'  END AS rating\_status  FROM restaurant r  NATURAL JOIN menu\_resto mr  NATURAL JOIN menu m  NATURAL JOIN order\_item i  WHERE m.menu\_name LIKE '%ayam%'  AND r.city = 'Bandung'; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | History, pesanan, order\_item, menu, menu\_resto, restaurant |

| **Informasi 12** | Deskripsi statistik dari menu: min harga, max harga, rerata harga, nama menu yang unik, nama kategori yang paling banyak beserta jumlahnya. |
| --- | --- |
| ***Query*** | CREATE VIEW statistik\_deskriptif\_menu AS  SELECT  MIN(menu\_price) AS min\_price,  MAX(menu\_price) AS max\_price,  AVG(menu\_price) AS avg\_price,  COUNT(DISTINCT menu\_name) AS unique\_menu\_count,    *--menghitung modus*  (SELECT menu\_name  FROM menu  GROUP BY menu\_name  ORDER BY COUNT(menu\_name) DESC  LIMIT 1) AS most\_frequent\_menu,  (SELECT COUNT(menu\_name)  FROM menu  GROUP BY menu\_name  ORDER BY COUNT(menu\_name) DESC  LIMIT 1) AS most\_frequent\_menu\_count  FROM menu; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | menu |

# 

| **Informasi 13** | Deskripsi statistik dari order\_address: jumlah nilai unik city, nilai modus city beserta frekuensinya, jumlah nilai unik subdistrik, nilai modus subdistrict beserta frekuensinya |
| --- | --- |
| ***Query*** | CREATE VIEW order\_address\_stats AS  SELECT  COUNT(DISTINCT city) AS jumlah\_kota\_unik,  (SELECT city FROM order\_address GROUP BY city ORDER BY  COUNT(\*) DESC LIMIT 1) AS kota\_modus,  (SELECT COUNT(\*) FROM order\_address WHERE city = (SELECT city FROM order\_address GROUP BY city ORDER BY COUNT(\*) DESC LIMIT 1)) AS frekuensi\_kota\_modus,  COUNT(DISTINCT subdistrict) AS jumlah\_subdistrict\_unik,  (SELECT subdistrict FROM order\_address GROUP BY subdistrict ORDER BY COUNT(\*) DESC LIMIT 1) AS subdistrict\_modus,  (SELECT COUNT(\*) FROM order\_address WHERE subdistrict = (SELECT subdistrict FROM order\_address GROUP BY subdistrict ORDER BY COUNT(\*) DESC LIMIT 1)) AS frekuensi\_subdistrict\_modus  FROM order\_address; |
| **Hasil *Query*** |  |
| **Tabel yang terlibat** | order\_address |

# 

# 

# Daftar Pustaka

Tim Pengajar IF2040 Semester I TA 2024/2025. (2024). *Relational Database Design Part 1.* Institut Teknologi Bandung.

Tim Pengajar IF2040 Semester I TA 2024/2025. (2024). *Relational Database Design Part 2.* Institut Teknologi Bandung.

Tim Pengajar IF2040 Semester I TA 2024/2025. (2024). *Relational Database Design Part 3.* Institut Teknologi Bandung.

Tim Pengajar IF2040 Semester I TA 2024/2025. (2024). *Relational Database Design Part 4.* Institut Teknologi Bandung.

MySQL Documentation. (n.d.). *MySQL 8.0 Reference Manual*. Diakses dari<https://dev.mysql.com/doc/refman/8.0/en/> pada 11 Desember 2024.

GeeksforGeeks. (n.d.). *SQL Tutorials*. Diakses dari <https://www.geeksforgeeks.org/sql-tutorial/> pada 11 Desember 2024.

# Pembagian Tugas

| **Nama Lengkap - NIM** | **Deskripsi Tugas** |
| --- | --- |
| Anggita Najmi Layali - 18223122 | * Menghubungi asisten dan menjadwalkan asistensi * Membantu revisi milestone 2 * Mengisi dokumen bagian Daftar *Functional Dependencies, Candidate Key*, dan Bentuk Normal relasi *Restaurant, Menu, Category* * Mengisi dokumen bagian Hasil Normalisasi Relasi *Menu* * Mengisi Daftar Informasi dari Basis Data: informasi 2, 3, 4, 5, 11, 12 |
| Muhammad Dzaky Atha F - 18223124 | * Membantu revisi milestone 2 * Merevisi file sql * Mengisi dokumen bagian Daftar *Functional Dependencies, Candidate Key*, dan Bentuk Normal relasi *user, user\_telp, order\_item, order\_address* * Mengisi dokumen bagian Daftar Informasi dari Basis Data: informasi 1, 10, 13 |
| M Rabbani K A - 18223130 | * Membantu revisi milestone 2 * Mengisi dokumen bagian Daftar *Functional Dependencies, Candidate Key*, dan Bentuk Normal relasi *History, Refund, Pesanan* * Mengisi dokumen bagian Daftar Informasi dari Basis Data: informasi 6, 7 |
| Sonya Putri Fadilah - 18223138 | * Membantu revisi milestone 2 * Mengisi dokumen bagian Daftar *Functional Dependencies, Candidate Key*, dan Bentuk Normal relasi *Shopeepay, Transaction, Driver* * Mengisi dokumen bagian Daftar Informasi dari Basis Data: informasi 8, 9 |

# Lampiran

## Moments of Meeting

**Asistensi Milestone 3**

| **Hari** | Kamis | **Tanggal** | 12 Desember 2024 |
| --- | --- | --- | --- |
| **Kelas** | 03 | **Kelompok** | 05 |
| **Waktu** | 20.30 - selesai | **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. | Apakah relasi/tabel baru hasil normalisasi perlu dicantumkan dalam relasional model? | Tidak perlu, relasional model pada bagian 2 hanya untuk revisi M2 saja. Jika ingin mencantumkan, buat ER-Model tersendiri. |
| 2. | Bonus statistik deskriptif buat semua relasi atau satu aja? | Tidak, boleh salah satu saja. |
| 3. | Mekanisme demo seperti apa? Apakah merupakan pengetesan query? | Karena berat di dokumen, tentunya dari segi dokumen juga akan diuji. |

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