

PROJECT 2 DATABASE SAINS DATA
DATABASE SISTEM ONLINE FOOD DELIVERY



Disusun Oleh:

Kelompok 3 Database Untuk Sains Data (B)

Annisa Fitria A.D.	(2106639900)
Kalisha Rahma Firza	(2106725053)
Kamal Muftie Yafi	(2106725034)
Nurkhalisha Humaira	(2106703802)
Rifqi Hafizuddin	(2106638204)
Vesya Padmadewi	(2106726926)

Dosen Pengampu:

Gianinna Ardanewari, M.Si.

PROGRAM STUDI S1 STATISTIKA
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
DEPOK
2023

DAFTAR ISI

BAB 1 PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Tujuan.....	4
BAB 2 METODE PENELITIAN.....	5
2.1 Jenis Metode Penelitian.....	5
2.2 Langkah Kerja.....	5
2.3 Software yang Digunakan.....	6
BAB 3 PEMBAHASAN.....	7
3.1 Entitas, Atribut, dan <i>Primary Key</i>	7
3.2 ER Diagram.....	9
3.3 Tabel Relasional.....	11
3.4 Simulasi.....	12
3.5 Analisis.....	19
BAB 4 PENUTUP.....	20
4.1 Kesimpulan.....	20
BAB 5 DAFTAR PUSTAKA.....	21

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Revolusi industri membawa perubahan besar dalam kehidupan manusia. Dampaknya menyebabkan teknologi dan sistem informasi berkembang dengan pesat. Metode-metode kuno yang cenderung lamban, tergantikan dengan metode modern yang singkat dan cepat. Adanya pergantian tersebut menandakan bahwa perkembangan teknologi dan sistem informasi zaman sekarang, telah meningkatkan efisiensi waktu serta produktivitas manusia dalam kehidupan sehari-hari.

Salah satu inovasi dari perkembangan teknologi dan informasi pada zaman modern ini adalah munculnya sistem pengiriman makanan atau yang biasa kita kenal dengan istilah *food delivery*. Pemanfaatan teknologi informasi pada sistem *food delivery* seperti aplikasi *mobile*, *website*, dan sistem manajemen pesanan, memungkinkan konsumen untuk memesan makanan dengan lebih mudah dan cepat, serta memungkinkan restoran dan pengirim makanan untuk memproses dan melacak pesanan dengan lebih efisien.

Adanya layanan *food delivery* disesuaikan dengan gaya hidup masyarakat di era modern yang tingkat produktivitasnya tinggi sehingga cenderung sibuk dan tidak memiliki banyak waktu untuk memasak. Meningkatnya permintaan akan pelayanan makanan cepat saji juga menjadi salah satu faktor munculnya sistem ini. Tentunya, *food delivery* memberikan kemudahan dan kenyamanan bagi konsumen untuk mendapatkan makanan tanpa harus datang ke restoran.

Hal inilah yang membuat *food delivery* menjadi semakin populer dan banyak digunakan saat ini. Tidak hanya restoran besar, tetapi juga restoran kecil, bahkan individu yang memiliki kemampuan memasak dapat memanfaatkan layanan *food delivery* untuk memperluas jangkauan pasar dan meningkatkan pendapatan. Banyaknya keunggulan dan manfaat dari *food delivery* membuat kami ingin meneliti mengenai sistem lebih lanjut di dalamnya.

1.2 Tujuan

1.2.1 Tujuan dari penelitian ini adalah:

1. Mengetahui database sistem yang mengatur layanan *food delivery*.
2. Menilik informasi lebih lanjut terkait layanan *food delivery*.
3. Mampu menjalankan relasi antar setiap data yang mempunyai keterkaitan dengan berdasarkan objek yang dihubungkan dengan suatu relasi khusus.

1.2.2 Tujuan dari adanya layanan *food delivery* adalah:

1. Dengan adanya layanan *food delivery*, konsumen dapat dengan mudah memesan makanan dari berbagai restoran tanpa harus keluar rumah atau kantor.
2. Dengan *food delivery*, konsumen dapat menghemat waktu yang sebelumnya digunakan untuk pergi ke restoran.
3. Dengan *food delivery*, konsumen memiliki akses ke berbagai restoran dan menu yang lebih luas.
4. Terdapat beberapa layanan *food delivery* yang menawarkan harga yang terjangkau dan bahkan menawarkan diskon atau promo untuk konsumen yang memesan makanan secara *online*.
5. Bagi restoran, *food delivery* adalah cara untuk memperluas pasar mereka dan menjangkau konsumen yang tidak dapat datang langsung ke restoran. Ini juga dapat meningkatkan pendapatan mereka dengan memungkinkan mereka untuk melayani lebih banyak pelanggan.

BAB 2

METODE PENELITIAN

2.1 Jenis Metode Penelitian

Metode penelitian yang kami gunakan untuk merancang *database* adalah dengan melakukan perancangan *database* level konseptual dan perancangan *database* level logika. Dengan kedua metode tersebut, kami dapat mengecek kebutuhan pengguna, batasan-batasan, dan hubungannya serta memetakan rancangan konseptual ke dalam model *database* yang akan digunakan.

2.2 Langkah Kerja

Berikut langkah kerja kelompok kami dalam membuat *database* sistem layanan *food delivery*:

- Menentukan tema untuk dibuat *database* sistemnya. Kami memilih untuk membuat *database* sistem dari layanan *food delivery* pada proyek ini.
- Berikut kami siapkan data untuk dimasukkan ke dalam *database*:

(insert gambar data dummy)

- Merancang *Database* Level Konseptual

Langkah-langkah untuk merancang *database* level konseptual:

- a) Membuat entitas dan atribut;
- b) Menentukan *primary key* untuk setiap entitas;
- c) Merancang sistem yang berlaku;
- d) Membuat ER Diagram.

- Merancang *Database* Level Logika

Langkah-langkah untuk merancang *database* level logika:

- a) Mengonversi ER Diagram ke dalam tabel;

- b) Melengkapi *record database* menggunakan data *dummy*;
- c) Menuliskan *code* dan melakukan simulasi CRUD dan join tabel;
- d) Menganalisis hasil dan membuat kesimpulan.

2.3 Software yang Digunakan

Software yang kami gunakan dalam merancang *database* sistem layanan *food delivery* ini meliputi:

1. Jupyter

Software ini digunakan untuk melakukan simulasi aplikasi layanan *food delivery* dengan menggunakan bahasa pemrograman Python.

2. DB Browser SQLite

Software ini digunakan untuk membentuk *database* sistem layanan *food delivery* yaitu dengan melakukan *create table*, *insert into* untuk memasukkan entitas, *read table* untuk memunculkan hasil tabel, *update table* untuk melakukan *update* data pada tabel, dan *delete table* untuk menghapus data pada tabel.

3. draw.io

Software ini digunakan untuk membentuk ER Diagram dan Tabel Relasional.

BAB 3

PEMBAHASAN

3.1 Entitas, Atribut, dan *Primary Key*

Dalam *database* sistem layanan *online food delivery* terdapat sejumlah entitas, yaitu; (1) pengguna, (2) pesanan, (3) pengiriman, (4) kurir, (5) restoran, dan (6) menu. Masing-masing entitas memiliki atribut sebagai berikut:

1. Pengguna

Entitas pengguna terdiri dari atribut yang berkaitan dengan informasi pengguna, yaitu:

- id_pengguna: ID unik pengguna (*Primary key*)
- nama_pengguna: Nama lengkap pengguna
- no_telp: Nomor telepon pengguna
- email: Alamat email pengguna
- alamat: Alamat pengiriman pengguna

2. Pesanan

Entitas pesanan terdiri dari atribut yang berkaitan dengan informasi pesanan, yaitu:

- no_order: Nomor unik order (*Primary key*)
- total_harga: Total harga pesanan
- waktu_pemesanan: Waktu pemesanan oleh pengguna
- status: Status pesanan
- id_pengguna: ID unik pengguna (*Foreign key*)
- id_produk: ID unik produk (*Foreign key*)
- id_pengiriman: ID unik pengiriman (*Foreign key*)

3. Pengiriman

Entitas pengiriman terdiri dari atribut yang berkaitan dengan informasi pengiriman, yaitu:

- id_pengiriman: ID unik pengiriman (*Primary key*)
- ongkos_pengiriman: Biaya untuk pengiriman

- waktu_pengiriman: Waktu yang dibutuhkan untuk pengiriman
- id_kurir: ID unik kurir (*Foreign key*)

4. Kurir

Entitas kurir terdiri dari atribut yang berkaitan dengan informasi kurir, yaitu:

- id_kurir: ID unik kurir (*Primary key*)
- nama_kurir: Nama kurir
- plat_kendaraan: Plat kendaraan kurir

5. Restoran

Entitas restoran terdiri dari atribut yang berkaitan dengan informasi restoran, yaitu:

- id_restoran: ID unik restoran (*Primary key*)
- nama_restoran: Nama restoran
- alamat: Alamat lengkap restoran
- rating: *Rating* restoran di aplikasi
- kontak: Kontak restoran yang dapat dihubungi

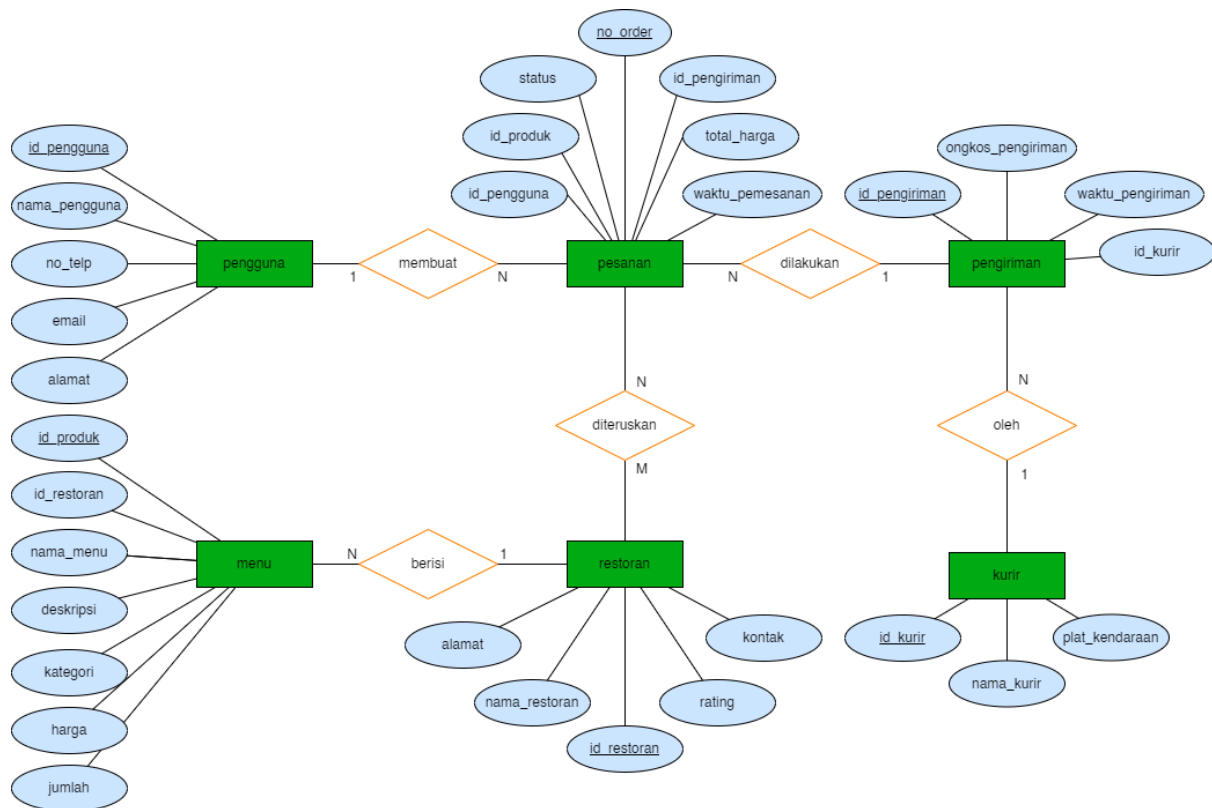
6. Menu

Entitas menu terdiri dari atribut yang berkaitan dengan informasi menu, yaitu:

- id_produk: ID unik produk (*Primary key*)
- id_restoran: ID unik restoran (*Foreign key*)
- nama_produk: Nama produk
- deskripsi: Deskripsi lengkap produk
- harga: Harga satu produk
- kategori: Kategori produk
- jumlah: Jumlah produk yang tersedia

3.2 ER Diagram

Berikut adalah ER Diagram yang kami buat dengan draw.io:



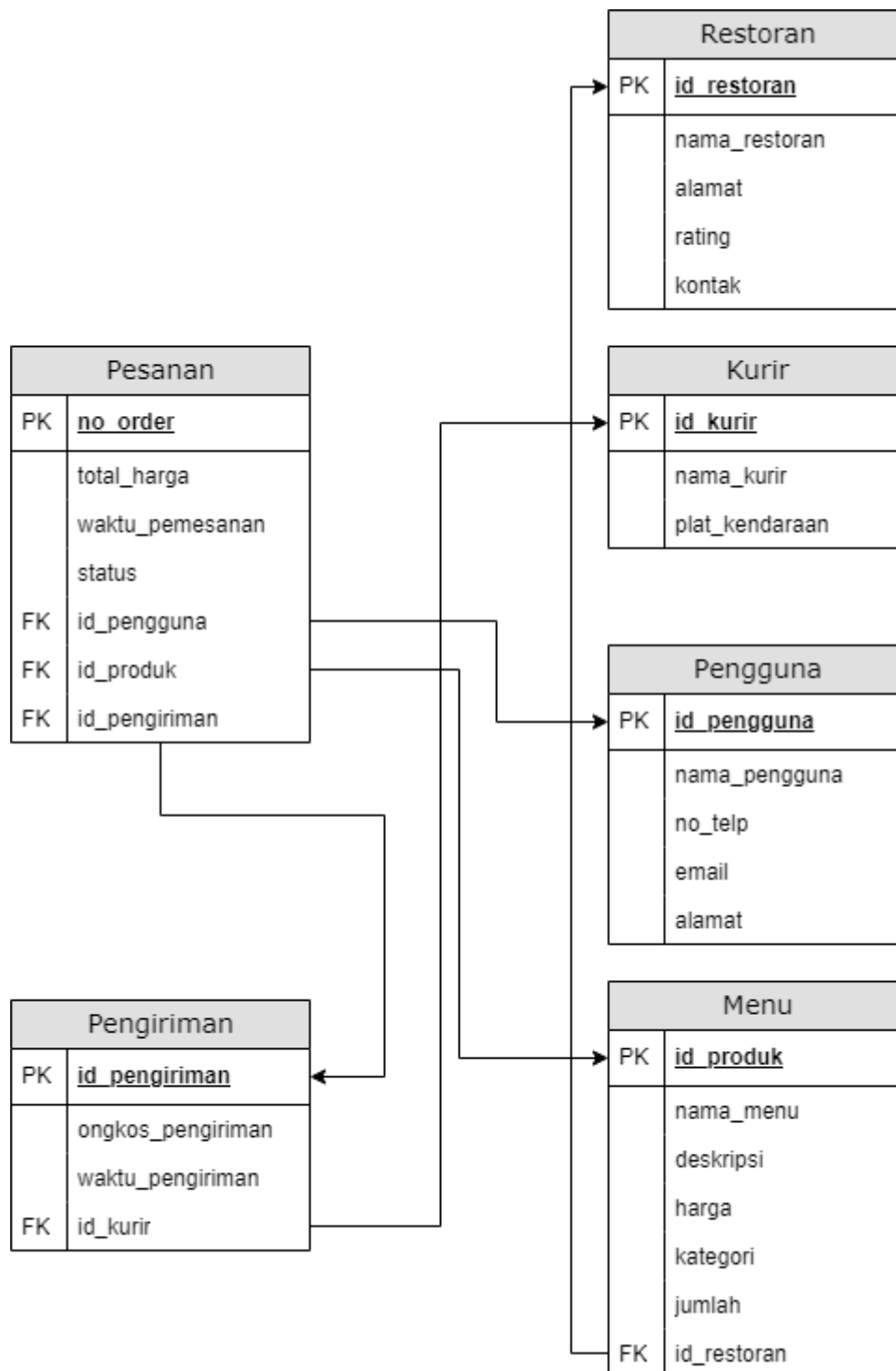
Penjelasan:

- Pengguna adalah sebuah *entity*, dengan *identifier* id_pengguna yang dibuat menjadi *primary key* yang digunakan untuk membedakan antarpengguna.
- Pesanan adalah sebuah *entity* dengan *identifier* no_order. Setiap pengguna dapat membuat pesanan, sehingga *entity* pengguna *partially participated* dalam hubungan *one-to-many* dengan *entity* pesanan, dihubungkan dengan id_pengguna pada *entity* pesanan.
- Restoran adalah sebuah *entity*, dengan *identifier* id_restoran. Setiap pesanan dapat diteruskan ke banyak restoran, sehingga *entity* pesanan *partially participated* dalam hubungan *many-to-many* dengan *entity* restoran, dihubungkan dengan id_restoran pada *entity* pesanan.
- Menu adalah sebuah *entity* dengan *identifier* id_produk. Restoran dan Menu terkait melalui *relationship one-to-many*; suatu restoran dapat memiliki banyak menu, dan

banyak menu dimiliki dalam satu restoran, dihubungkan dengan id_restoran pada *entity* menu.

- Pengiriman adalah sebuah *entity* dengan *identifier* id_pengiriman. Setiap pesanan dilakukan dalam satu pengiriman, sehingga *entity* pesanan *partially participated* dalam hubungan *many-to-one* dengan *entity* pengiriman, dihubungkan dengan id_pengiriman pada *entity* pesanan.
- Kurir adalah sebuah *entity* dengan *identifier* id_kurir. Banyak pengiriman dilakukan oleh setiap kurir; pengiriman dan kurir terkait dalam hubungan *many-to-one*, dihubungkan dengan id_kurir pada *entity* pengiriman.

3.3 Tabel Relasional



Seorang Pengguna memiliki id_pengguna (primary key), nama_pengguna, no_telp, email, dan alamat. Seorang kurir memiliki id_kurir (primary key), nama_kurir, dan plat_kendaraan. Sebuah restoran memiliki id_restoran (primary key), nama_restoran, alamat, rating, dan kontak. Setiap restoran memiliki menu dengan id_produk (primary key),

nama_menu, deskripsi, harga, kategori, jumlah, dan id_restoran yang bergantung pada restoran asal menu tersebut.

Pesanan memiliki no_order, total_harga, waktu_pemesanan, dan status. Pengiriman memiliki id_pengiriman (primary_key), ongkos_pengiriman, dan waktu_pengiriman. Suatu pesanan berupa produk dari menu sebuah restoran yang dipesan oleh seorang pengguna dan dikirim oleh seorang kurir.

3.4 Simulasi

Berikut adalah tautan untuk simulasi Database kami:

<https://drive.google.com/drive/folders/185EccIWfpu8TjmqS9QSdBY2c3Bx3r1x?usp=sharing>

3.4.1 Create Table

```
1 CREATE TABLE Kurir (  
2     id_kurir INTEGER NOT NULL UNIQUE,  
3     nama_kurir VARCHAR(50) NOT NULL,  
4     plat_kendaraan VARCHAR(10) NOT NULL,  
5     PRIMARY KEY(id_kurir)  
6 )
```

```
1 CREATE TABLE Menu (  
2     id_produk INTEGER NOT NULL UNIQUE,  
3     id_restoran INTEGER NOT NULL,  
4     nama_produk VARCHAR(50) NOT NULL,  
5     deskripsi VARCHAR(100),  
6     harga FLOAT NOT NULL,  
7     kategori VARCHAR(20) NOT NULL,  
8     jumlah INTEGER,  
9     PRIMARY KEY(id_produk),  
10    FOREIGN KEY(id_restoran) REFERENCES Restoran(id_restoran)  
11 )
```

```
1 CREATE TABLE Pengguna (  
2     id_pengguna INTEGER NOT NULL UNIQUE,  
3     nama_pengguna VARCHAR(50) NOT NULL,  
4     no_telp INTEGER NOT NULL UNIQUE,  
5     email VARCHAR(30) NOT NULL UNIQUE,  
6     alamat VARCHAR(100) NOT NULL,  
7     PRIMARY KEY(id_pengguna)  
8 )
```

```
1 CREATE TABLE Pengiriman (  
2     id_pengiriman INTEGER NOT NULL UNIQUE,  
3     ongkos_pengiriman FLOAT,  
4     waktu_pengiriman DATETIME,  
5     id_kurir INTEGER,  
6     PRIMARY KEY(id_pengiriman),  
7     FOREIGN KEY(id_kurir) REFERENCES Kurir(id_kurir)  
8 )
```

```

1 CREATE TABLE Pesanan (
2     no_order INTEGER NOT NULL UNIQUE,
3     total_harga FLOAT NOT NULL,
4     waktu_pemesanan DATETIME NOT NULL,
5     status VARCHAR(20) NOT NULL,
6     id_pengguna INTEGER,
7     id_produk INTEGER,
8     id_pengiriman INTEGER,
9     PRIMARY KEY(no_order),
10    FOREIGN KEY(id_pengguna) REFERENCES Pengguna(id_pengguna),
11    FOREIGN KEY(id_pengiriman) REFERENCES Pengiriman(id_pengiriman),
12    FOREIGN KEY(id_produk) REFERENCES Menu(id_produk)
13 )

```

```

1 CREATE TABLE Restoran (
2     id_restoran INTEGER NOT NULL UNIQUE,
3     nama_restoran VARCHAR(50),
4     alamat VARCHAR(100),
5     rating FLOAT,
6     kontak VARCHAR(30),
7     PRIMARY KEY(id_restoran)
8 )

```

3.4.2 Insert Into

```

1 INSERT INTO Kurir
2 VALUES
3 (10101, 'Rochmat', 'B 1234 CDE'),
4 (10102, 'Agus', 'B 1357 EFG'),
5 (10103, 'Rio', 'B 2468 GHI'),
6 (10104, 'Budi', 'B 8247 UJK'),
7 (10105, 'Yanto', 'F 9462 AKP')

```

```

1 INSERT INTO Menu
2 VALUES
3 (20201, 60601, 'Nasi Cumi Item', 'Nasi, Cumi Item, Mie, Serundeng, Peyek Udang, Sambal Pencit', 45000, 'Makanan', 100),
4 (20202, 60602, 'Mie Gacoan Lvl', 'Mie kenyal, Cabai, Ayam suwir, Pangsit goreng', 14500, 'Makanan', 50),
5 (20203, 60603, 'Coconut Jelly Milk Tea', 'Milk tea dengan topping coconut jelly yang lembut', 21000, 'Minuman', 10),
6 (20204, 60604, 'Ayam Goreng Paha', 'Nasi + Ayam Goreng + Sambal + Lalapan', 24000, 'Makanan', 36),
7 (20205, 60605, 'Combo A', 'Papas pizza large, Mac n cheese, Garlic bread, 2 Lemon tea', 16000, 'Makanan', 5)

```

```

1 INSERT INTO Pengguna
2 VALUES
3 (30301, 'Casia', '08123443210', 'casia01@gmail.com', 'Jl. Meruya Ilir No. 25, Jakarta Barat'),
4 (30302, 'Zhafira', '087865780099', 'zhafiraa30@gmail.com', 'Jl. Ampera Raya No.11, Jakarta Selatan'),
5 (30303, 'Aisha', '085866563443', 'aishazhar@gmail.com', 'Jl. Asia Afrika No. 2, Jakarta Pusat'),
6 (30304, 'Calvin', '087899006756', 'calvinhar@gmail.com', 'Jl. Bougenville Blok C5, Bekasi'),
7 (30305, 'Aksa', '085670093456', 'aksaputra@gmail.com', 'Jl. Kelapa Nias Blok A3, Jakarta Utara')

```

```

1 INSERT INTO Pengiriman
2 VALUES
3 (40401, 20000, '35 menit', 10101),
4 (40402, 17000, '30 menit', 10102),
5 (40403, 10000, '15 menit', 10103),
6 (40404, 19000, '20 menit', 10104),
7 (40405, 12000, '12 menit', 10105)

```

```

1 INSERT INTO Pesanan
2 VALUES
3 (50501, 65000, '2023-06-14 17:34:11', 'Disiapkan', 30301, 20201, 40401),
4 (50502, 70000, '2023-06-14 20:21:12', 'Dikirim', 30302, 20202, 40402),
5 (50503, 53000, '2023-06-14 22:52:33', 'Dikirim', 30303, 20203, 40403),
6 (50504, 44000, '2023-06-15 07:18:46', 'Disiapkan', 30304, 20204, 40404),
7 (50505, 39000, '2023-06-15 09:26:28', 'Dikirim', 30305, 20205, 40405)

```

```

1 INSERT INTO Restoran
2 VALUES
3 (60601, 'Nasi Cumi Hitam Madura Pak Kris', 'Jl. H. Nawi Raya No.28, Jakarta Selatan', '4.4', '082310107778'),
4 (60602, 'Mie Gacoan', 'Jl. Bintara, Kota Bekasi', '4.5', '0895414854300'),
5 (60603, 'Mixue', 'Jl. Sumur Batu Raya No.14, Jakarta Pusat', '4.6', '08118160829'),
6 (60604, 'Ayam Gepuk Pak Gembus', 'Jl. Meruya Ilir Raya No. 88, Jakarta Barat', '4.7', '08161998777'),
7 (60605, 'Papa Rons Pizza', 'Jl. Kelapa Nias Blok HF3, Jakarta Utara', '4.3', '085810616466')

```

3.4.3 Read

Pada tahap ini, akan dilakukan simulasi untuk membaca dan melihat data pada tabel (read). Berikut adalah simulasinya

Misalkan ingin diketahui daftar pesanan pengguna dengan status pengiriman, alamat pengiriman, produk yang dipesan, dan jumlah yang harus dibayar. Maka dapat dilakukan dengan join tabel pengguna dengan menu dan pesanan.

```
SELECT Pesanan.no_order, Pengguna.nama_pengguna, Pengguna.alamat, Menu.nama_produk, Pesanan.status, Pesanan.total_harga
FROM Pengguna JOIN Menu JOIN Pesanan
on Pengguna.id_pengguna=Pesanan.id_pengguna AND Pesanan.id_produk=Menu.id_produk
```

	no_order	nama_pengguna	alamat	nama_produk	status	total_harga
1	50501	Casia	Jl. Meruya Ilir No. 25, Jakarta Barat	Nasi Cumi Item	Disiapkan	65000.0
2	50502	Zhafira	Jl. Ampera Raya No.11, Jakarta Selatan	Mie Gacoan LV1	Dikirim	70000.0
3	50503	Aisha	Jl. Asia Afrika No. 2, Jakarta Pusat	Coconut Jelly Milk Tea	Dikirim	53000.0
4	50504	Calvin	Jl. Bougenville Blok C5, Bekasi	Ayam Goreng Paha	Disiapkan	44000.0
5	50505	Aksa	Jl. Kelapa Nias Blok A3, Jakarta Utara	Combo A	Dikirim	39000.0

Misalkan ingin diketahui daftar restoran dengan alamat dan kontak restoran tersebut, serta nama, deskripsi, kategori, dan harga produk yang dapat dipesan dari setiap restoran. Maka dapat dilakukan dengan join tabel restoran dengan menu

```
SELECT Restoran.nama_restoran, Restoran.alamat, Restoran.kontak, Menu.nama_produk, Menu.deskripsi, Menu.kategori, Menu.harga
FROM Restoran JOIN Menu
on Restoran.id_restoran=Menu.id_restoran
```

	nama_restoran	alamat	kontak	nama_produk	deskripsi	kategori	harga
1	Nasi Cumi Hitam Madura Pak Kris	Jl. H. Nawi Raya No.28, Jakarta Selatan	082310107778	Nasi Cumi Item	Nasi, Cumi Item, Mie, Serundeng, ...	Makanan	45000.0
2	Mie Gacoan	Jl. Bintara, Kota Bekasi	0895414854300	Mie Gacoan LV1	Mie kenyal, Cabai, Ayam suwir, Pang...	Makanan	14500.0
3	Mixue	Jl. Sumur Batu Raya No.14, Jakarta ...	08118160829	Coconut Jelly Milk Tea	Milk tea dengan topping coconut jelly...	Minuman	21000.0
4	Ayam Gepuk Pak Gembus	Jl. Meruya Ilir Raya No. 88, Jakarta ...	08161998777	Ayam Goreng Paha	Nasi + Ayam Goreng + Sambal + ...	Makanan	24000.0
5	Papa Rons Pizza	Jl. Kelapa Nias Blok HF3, Jakarta Utara	085810616466	Combo A	Papas pizza large, Mac n cheese, Gar...	Makanan	16000.0

Misalkan ingin diketahui id, nama, dan plat kendaraan kurir serta waktu dan ongkos pengiriman. Maka dapat dilakukan dengan join tabel kurir dengan pengiriman.

```
SELECT Kurir.id_kurir, Kurir.nama_kurir, Kurir.plat_kendaraan, Pengiriman.waktu_pengiriman, Pengiriman.ongkos_pengiriman
FROM Kurir JOIN Pengiriman
on Kurir.id_kurir=Pengiriman.id_kurir
```

	id_kurir	nama_kurir	plat_kendaraan	waktu_pengiriman	ongkos_pengiriman
1	10101	Rochmat	B 1234 CDE	35 menit	20000.0
2	10102	Agus	B 1357 EFG	30 menit	17000.0
3	10103	Rio	B 2468 GHI	15 menit	10000.0
4	10104	Budi	B 8247 UJK	20 menit	19000.0
5	10105	Yanto	F 9462 AKP	12 menit	12000.0

3.4.4 Update

Pada tahap ini, akan dilakukan simulasi untuk memperbarui data pada tabel (Update). Berikut adalah simulasinya.

Misalkan ingin dilakukan update mengenai status pesanan bahwa pesanan telah selesai. Maka, perintah *query* yang digunakan serta *output* yang dihasilkan adalah sebagai berikut.

```
1 UPDATE "Pesanan" SET "status" = "Selesai" WHERE "no_order" = "50501"
```

```
1 UPDATE "Pesanan" SET "status" = "Selesai" WHERE "no_order" = "50502"
```

```
1 UPDATE "Pesanan" SET "status" = "Selesai" WHERE "no_order" = "50503"
```

```
1 UPDATE "Pesanan" SET "status" = "Selesai" WHERE "no_order" = "50504"
```

```
1 UPDATE "Pesanan" SET "status" = "Selesai" WHERE "no_order" = "50505"
```

	no_order	total_harga	waktu_pemesanan	status	id_pengguna	id_produk	id_pengiriman
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	50501	65000.0	2023-06-14 17:34:11	Selesai	30301	20201	40401
2	50502	70000.0	2023-06-14 20:21:12	Selesai	30302	20202	40402
3	50503	53000.0	2023-06-14 22:52:33	Selesai	30303	20203	40403
4	50504	44000.0	2023-06-15 07:18:46	Selesai	30304	20204	40404

Setelah tabel pesanan di-update, akan dilakukan *read table* untuk menampilkan pengguna dengan harga produk yang dipesannya. Maka tabel yang didapat sebagai berikut.

	no_order	nama_pengguna	alamat	nama_produk	status	total_harga
1	50501	Casia	Jl. Meruya Ilir No. 25, Jakarta Barat	Nasi Cumi Item	Selesai	65000.0
2	50502	Zhafira	Jl. Ampera Raya No.11, Jakarta Selatan	Mie Gacoan LV1	Selesai	70000.0
3	50503	Aisha	Jl. Asia Afrika No. 2, Jakarta Pusat	Coconut Jelly Milk Tea	Selesai	53000.0
4	50504	Calvin	Jl. Bougenville Blok C5, Bekasi	Ayam Goreng Paha	Selesai	44000.0
5	50505	Aksa	Jl. Kelapa Nias Blok A3, Jakarta Utara	Combo A	Selesai	39000.0

3.4.5 Delete

Pada tahap ini, akan dilakukan simulasi untuk menghapus record pada tabel (Delete). Berikut adalah simulasinya.

Misalkan Restoran Papa Rons Pizza mengadakan menu promo pada acara *Anniversary* satu tahun restorannya, yaitu menu Combo A. Namun saat ini masa promo tersebut sudah habis, maka menu promo tersebut akan dihapus. Maka, perintah *query* yang digunakan serta *output* yang dihasilkan adalah sebagai berikut.

```
1 DELETE FROM Menu WHERE id_produk = 20205
```

	id_produk	id_restoran	nama_produk	deskripsi	harga	kategori	jumlah
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	20201	60601	Nasi Cumi Item	Nasi, Cumi Item, Mie, Serundeng, ...	45000.0	Makanan	100
2	20202	60602	Mie Gacoan LV1	Mie kenyal, Cabai, Ayam suwir, Pang...	14500.0	Makanan	50
3	20203	60603	Coconut Jelly Milk Tea	Milk tea dengan topping coconut jelly...	21000.0	Minuman	10
4	20204	60604	Ayam Goreng Paha	Nasi + Ayam Goreng + Sambal + ...	24000.0	Makanan	36

3.4.6 Aplikasi

Berikut adalah aplikasi yang kami buat sebagai simulasi penggunaan *database management system*:


```

1 import sqlite3
2 import pandas as pd
3 import random as rd
4 import datetime
5 from datetime import datetime, timedelta, date
6 import numpy as np
7
8 conn = sqlite3.connect('fooddelivery.db')
9 c = conn.cursor()
10
11 insertkurir = 'INSERT INTO Kurir VALUES (?,?,'
12 insertmenu = 'INSERT INTO Menu VALUES (?,?,'
13 insertpengguna = 'INSERT INTO Pengguna VALUES (?,?,'
14 insertpengiriman = 'INSERT INTO Pengiriman VALUES (?,?,'
15 insertpesanan = 'INSERT INTO Pesanan VALUES (?,?,'
16 insertrestoran = 'INSERT INTO Restoran VALUES (?,?,'
17
18 while True:
19     print(pd.read_sql_query('SELECT * FROM Restoran', conn))
20     posisi = int(input('\nApa tujuan Anda di sini? (Seller (0) / Customer (1) / Driver (2)): '))
21     if posisi == 1:
22         print("Apakah Anda ingin membeli Makanan/Minuman?")
23         choice = int(input("Ingin membeli barang (1) / Tidak ingin membeli barang (0): "))
24         if choice == 0:
25             print("Terima kasih sudah melihat Makanan/Minuman di aplikasi kami!")
26             break
27         else:
28             uid = rd.randint(30300,30399)
29             print("Silakan masukkan data diri Anda!")
30             uname = str(input("Masukkan nama Anda: "))
31             uphone = int(input("Masukkan nomor telepon Anda: "))
32             umail = str(input("Masukkan alamat email Anda: "))
33             uaddress = str(input("Masukkan alamat pengiriman Anda: "))
34             c.execute(insertpengguna, (uid, uname, uphone, umail, uaddress))
35
36             resto = int(input("Masukkan id_restoran yang ingin Anda lihat menyanya: "))
37             print(pd.read_sql_query('SELECT * FROM Menu', conn))
38
39             menu = int(input("Masukkan id_menu yang ingin Anda beli: "))
40             qty = int(input("Masukkan jumlah yang ingin Anda beli: "))
41             currentTimeDate = datetime.now()
42             print("Durasi Pengiriman:\n30 menit: Rp10.000,00\n1 jam: Rp15.000,0")
43             durasi = str(input("Masukkan lama pengiriman (30 menit (0) / 1 jam (1)): "))
44             oid = rd.randint(50500,50599)
45             if durasi == 0:
46                 sid = rd.randint(40400,40499)
47                 cid = rd.randint(10100,10199)
48                 c.execute(insertpengiriman, (sid, 10000, '30 menit', cid))
49             else:
50                 sid = rd.randint(40400,40499)
51                 cid = rd.randint(10100,10199)
52                 c.execute(insertpengiriman, (sid, 15000, '1 jam', cid))
53             totprice = int(input("Masukkan total biaya Anda: "))
54             c.execute(insertpesanan, (oid, totprice, currentTimeDate, 'Disiapkan', uid, menu, sid))
55             print("Terima kasih sudah melakukan pembelian di aplikasi Kami!")
56             break
57     elif posisi == 0:
58         rid = rd.randint(60600,60699)
59         rname = str(input("Masukkan nama Restoran Anda: "))
60         raddress = str(input("Masukkan alamat Restoran Anda: "))
61         rrating = float(input("Masukkan rating Restoran Anda: "))
62         rphone = str(input("Masukkan nomor telepon Restoran Anda: "))
63         c.execute(insertrestoran, (rid, rname, raddress, rrating, rphone))
64
65         mid = rd.randint(20200,20299)
66         mname = str(input("Masukkan nama menu yang ingin Anda jual: "))
67         mdesc = str(input("Masukkan deskripsi dari menu yang ingin Anda jual: "))
68         mprice = int(input("Masukkan harga dari menu yang ingin Anda jual: "))
69         mcat = str(input("Masukkan kategori dari menu yang ingin Anda jual: "))
70         mqty = str(input("Masukkan jumlah dari menu yang ingin Anda jual: "))
71         c.execute(insertmenu, (mid, rid, mname, mdesc, mprice, mcat, mqty))
72
73         print("Terima kasih menjual masakan Anda di aplikasi Kami!")
74         break
75     else:
76         cid = int(input("Masukkan id_kurir Anda: "))
77         print(pd.read_sql_query('SELECT * FROM Pesanan', conn))
78         break

```

Pada awal aplikasi, akan ditanyakan kepada pengakses aplikasi, apa tujuannya di sini? Apakah sebagai (0) Seller, (1) Customer, atau (2) Driver dan ditampilkan daftar restoran.

- Seller

Akan di-random id_restoran sebagai identifier restoran dari seller. Selanjutnya, seller akan diminta untuk mengisi beberapa data restoran, seperti nama restoran, alamat restoran, rating, dan nomor telepon restoran. Kemudian, data-data tersebut dimasukkan ke dalam tabel Restoran pada database.

Setelah membuat restoran, seller juga diminta untuk memasukkan menu yang ingin dijual. Dengan tahapan *random* id_menu sebagai *identifier* menu dan memasukkan data menu, seperti nama menu, deskripsi menu, harga menu, kategori menu, dan jumlah menu tersebut yang dijual.

Terakhir, akan ditampilkan pesan “Terima kasih sudah menjual masakan di aplikasi kami”.

- Customer

Pengguna akan ditanyakan apakah ingin membeli makanan/minuman. Jika tidak, maka akan diberi ucapan “Terima kasih sudah melihat makanan/minuman di aplikasi kami”. Jika iya, akan di-random id_pengguna sebagai *identifier* pengguna dan pengguna akan diminta untuk mengisi data diri, seperti nama, nomor telepon, email, dan alamat pengiriman. Kemudian, data-data tersebut dimasukkan ke dalam tabel Pengguna pada database.

Setelah mengisi data diri pengguna, pengguna akan diminta untuk memilih menu apa yang ingin dibeli dalam bentuk id_menu dan jumlahnya. Akan diambil waktu saat ini dan pengguna diminta untuk memilih opsi pengiriman, apakah ingin pengiriman dalam 30 menit atau 1 jam. Kemudian, data-data tersebut akan dimasukkan ke dalam tabel pengiriman.

Terakhir, pengguna diminta untuk membayar dengan menuliskan total harga. Kemudian, data-data tersebut akan dimasukkan ke dalam tabel pesanan dan ditampilkan pesan “Terima kasih sudah melakukan pembelian di aplikasi kami”.

- Driver

Driver akan diminta untuk memasukkan id_kurir kemudian ditampilkan apa saja pesanan yang sedang berjalan.

3.5 Analisis

Pada tahap ini, akan dilakukan simulasi untuk menganalisis database dengan menggunakan join table. Misalkan terjadi kesalahan pesanan, sehingga ingin diketahui nomor pesanan, nama pengguna, dan nama kurir. Maka dapat dianalisis menggunakan join table. Karena tidak ada atribut yang menghubungkan tabel pesanan dan pengguna dengan tabel kurir, maka akan digunakan tabel perantara untuk menghubungkan semua tabel, yaitu tabel pengiriman, sehingga kita dapat join tabel pesanan dengan pengguna dan kurir dan pengiriman.

```
SELECT Pesanan.no_order, Pengguna.nama_pengguna, Kurir.nama_kurir, Pengiriman.id_kurir
FROM Pesanan JOIN Pengguna JOIN Kurir JOIN Pengiriman
on Pengguna.id_pengguna=Pesanan.id_pengguna AND Kurir.id_kurir=Pengiriman.id_kurir AND Pengiriman.id_pengiriman=Pesanan.id_pengiriman
```

	no_order	nama_pengguna	nama_kurir	id_kurir
1	50501	Casia	Rochmat	10101
2	50502	Zhafira	Agus	10102
3	50503	Aisha	Rio	10103
4	50504	Calvin	Budi	10104
5	50505	Aksa	Yanto	10105

Misalkan diketahui bahwa kesalahan pesanan dilakukan oleh kurir yang bernama Agus, maka kita dapat menggunakan kode yang sama seperti sebelumnya dan ditambahkan “WHERE kurir.nama_kurir=’Agus’ ”

```
SELECT Pesanan.no_order, Pengguna.nama_pengguna, Kurir.nama_kurir, Pengiriman.id_kurir
FROM Pesanan JOIN Pengguna JOIN Kurir JOIN Pengiriman
on Pengguna.id_pengguna=Pesanan.id_pengguna AND Kurir.id_kurir=Pengiriman.id_kurir AND Pengiriman.id_pengiriman=Pesanan.id_pengiriman
WHERE kurir.nama_kurir="Agus"
```

	no_order	nama_pengguna	nama_kurir	id_kurir
1	50502	Zhafira	Agus	10102

BAB 4

PENUTUP

4.1 Kesimpulan

Dengan menggunakan metode penelitian yang dipilih, telah dibentuk *Database food delivery* dalam mengelola sistem untuk pengiriman makanan yang mencakup tabel pesanan, pengiriman, restoran, kurir, pengguna, dan menu. *Database* ini terus digunakan dalam pencatatan informasi hingga manajemen data karena dapat meningkatkan efisiensi operasional dan memonitor kegiatan bisnis. Dengan memanfaatkan *database* ini, diharapkan pengelola dapat melakukan analisis data dengan lebih efisien dan akurat, meningkatkan kualitas keputusan yang diambil, serta menghadirkan perbaikan dalam penyediaan layanan.

BAB 5

DAFTAR PUSTAKA

- CNBC Indonesia. (2022, September 21). *Mengenal Apa Itu Internet, Sejarah Perkembangan & Manfaatnya*. CNBC Indonesia. Diakses pada 16 Juni 2023, dari <https://www.cnbcindonesia.com/tech/20220921131159-37-373856/mengenal-apa-itu-internet-sejarah-perkembangan-manfaatnya>
- Laruan. (2021, Juli 22). *7 Keuntungan dan Kerugian Menggunakan Food Delivery*. Kredit Pintar. Diakses pada 16 Juni 2023, dari <https://www.kreditpintar.com/education/7-keuntungan-dan-kerugian-menggunakan-food-delivery>
- Salamun, Zainal. (2016, September 8). *PELACAKAN LOKASI DENGAN GOOGLE MAP MENGGUNAKAN FITUR PERKIRAAN WAKTU SAMPAI (ETA) PADA APLIKASI DELIVERY PIZZA BERBASIS ANDROID*. Skripsi thesis, STMIK AKAKOM Yogyakarta. Diakses pada 16 Juni 2023, dari <http://eprints.akakom.ac.id/id/eprint/1604>