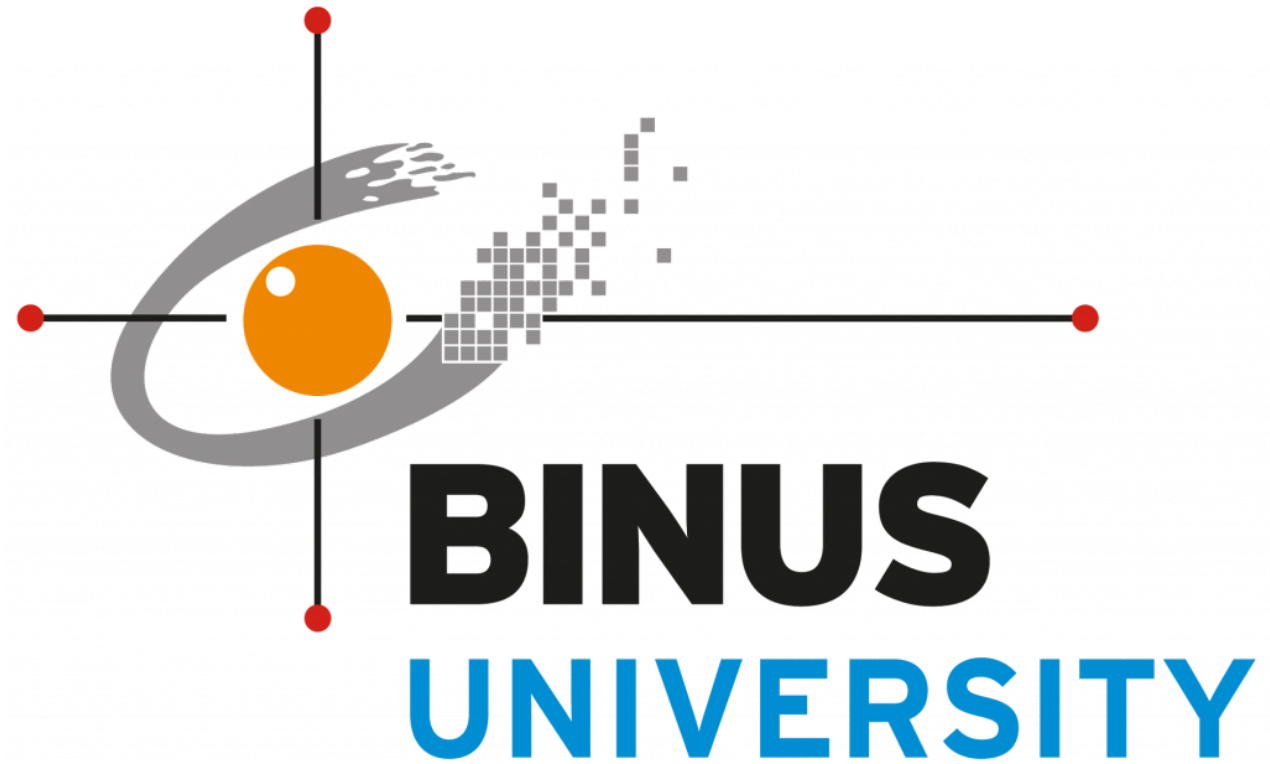


## Assurance of Learning Database Technology (COMP6799001)



Disusun Oleh [LA09] :

1. Agnes Calista - 2501980690
2. Belinda Mutiara - 2540119596
3. Caroline Angelina Sunarya - 2501995093
4. Florencia - 2540129262
5. Gabrielle Felicia Ariyanto - 2540134874
6. Jasmine Mutia Alifa - 2502026873

## Daftar Isi

<b>BAB I</b>	<b>3</b>
<b>PENDAHULUAN</b>	<b>3</b>
<b>BAB II</b>	<b>5</b>
<b>LANDASAN TEORI</b>	<b>5</b>
1. Functional Dependency	5
2. Unnormal Form (UNF)	5
3. First Normal Form (1NF)	5
4. Second Normal Form (2NF)	6
5. Third Normal Form (3NF)	6
6. Boyce-Codd Normal Form (BCNF)	7
7. Entity Relationship Diagram(ERD)	7
<b>BAB III</b>	<b>8</b>
<b>METODE PENELITIAN</b>	<b>8</b>
A. Tools	8
B. Material	8
C. Implementasi Metode	9
D. Prosedur Penelitian	10
E. Metode Mengumpulkan Data	10
<b>BAB IV</b>	<b>11</b>
<b>IMPLEMENTASI</b>	<b>11</b>
UNF (Unnormalized Form)	11
1NF (1st Normal Form)	13
2NF (2nd Normal Form)	16
3NF (3rd Normal Form)	18
BCNF (Boyce-Codd Normal Form)	22
Entity Relationship Diagram (ERD) in Crow's Foot Notation	26
<b>Link Gambar ERD:</b>	<b>27</b>
Source Code	27
Output	36
Struktur	39
<b>BAB V</b>	<b>40</b>
<b>KESIMPULAN</b>	<b>40</b>
<b>DAFTAR PUSTAKA</b>	<b>41</b>



# BAB I

## PENDAHULUAN

Data merupakan kumpulan fakta yang dapat direkam dan memiliki makna implisit. Data sudah menjadi bagian dalam kehidupan manusia yang tidak terpisahkan. Mulai dari nama lengkap, usia, alamat, jenis kelamin, nomor telepon, atau bahkan durasi tidur per hari pun dapat dicatat dan dimasukkan kedalam data.

*Database* adalah koleksi data yang saling terintegrasi dan sifatnya tersentralisasi. *Database* berisi kumpulan informasi yang disimpan secara sistematis pada suatu perangkat sehingga nantinya dapat diakses oleh sistem berupa *query*. *Database Management System* atau *DBMS* adalah perangkat lunak yang digunakan untuk menyimpan dan mengelola *database*. Hal ini dapat membantu pengguna untuk mengekstrak informasi yang berguna dengan memproses data terlebih dahulu di dalam *database* melalui beberapa tahap seperti mengubah data menjadi bentuk normal kemudian memodifikasinya menjadi model relasional yang terstruktur rapi, dimana relasi ditampilkan dengan jelas, mudah dipahami, dan dikendalikan. Terdapat beberapa perangkat lunak yang termasuk ke dalam *DBMS* misalnya *MS-SQL Server*, *MySQL*, *Microsoft-Access*, *FoxBase*, dan masih banyak lagi. Terdapat berbagai macam jenis *database* yang telah digunakan dalam kehidupan, contoh sederhananya ialah sistem penyimpanan data pribadi pada perangkat komputer. Selain itu terdapat juga jenis *database* yang dijalankan melalui internet, jenis ini disebut sebagai *cloud database*, *cloud database* biasa digunakan pada *google drive*, *one drive*, dan masih banyak lagi.

Data adalah dasar untuk mengambil keputusan, akan tetapi sebelum sampai di titik tersebut, data harus terlebih dahulu diolah. Alasan mengapa data perlu diolah terlebih dahulu ialah karena dengan adanya pengolahan data, pengguna dapat menghindari adanya redundansi data, data inkonsisten, ataupun anomali data. Anomali yang dimaksud ialah *insertion anomaly*, *modification anomaly*, serta *deletion anomaly*. *Insertion Anomaly* adalah proses penambahan data baru yang mempengaruhi user untuk terjadinya duplikasi data. *Modification Anomaly* adalah proses pengubahan data yang akan mempengaruhi perubahan record lain karena adanya duplikasi. *Deletion Anomaly* adalah penghapusan baris data yang menyebabkan hilangnya data pada record lain.

Langkah dalam mengolah data ialah pertama sebelum data disimpan ke dalam

*DBMS*, kita perlu memahami dengan jelas jenis data apa yang sedang kita kerjakan, apakah ada relasi dalam data tersebut, apakah datanya bersih, apakah ada anomali dan sebagainya. Langkah selanjutnya ialah melakukan normalisasi data mulai dari menentukan *functional dependency* hingga membuat data kedalam bentuk *Boyce-Codd Normal Form (BCNF)*. Langkah ketiga ialah membuat skema relasional menjadi *ERD (Entity Relationship Diagram)* agar relasinya jelas untuk dilihat dan mudah dipahami.

Pada kesempatan kali ini, penulis menggunakan struk transaksi pembelian makanan melalui *e-commerce*, yakni *go-food* seperti yang sudah terlampir pada lampiran. Struk berisi berbagai macam atribut yang nantinya akan penulis teliti dan olah menjadi database serta dinormalisasikan hingga menjadi ERD.

## BAB II

### LANDASAN TEORI

Untuk melakukan normalisasi pada database, kami menerapkan beberapa langkah, mulai dari mencari *functional dependency*, membentuk *un-normal form*, *first normal form*, *second normal form*, *third normal form*, *Boyce-Codd normal form*, hingga ERD. Berikut adalah penjelasan lebih lengkap pada masing-masing langkah yang kami lakukan:

#### 1. Functional Dependency

*Functional dependency* merupakan gambaran yang menggambarkan hubungan ketergantungan antar 2 set atau lebih atribut dalam suatu *database*. *Functional dependency* biasanya dituliskan dalam bentuk  $X \rightarrow Y$  dimana X dan Y adalah subset dari kumpulan atribut dalam suatu row. Nilai Y ditentukan dari nilai X dan nilai X secara unik atau secara fungsional menentukan nilai Y.

#### 2. Unnormal Form (UNF)

*Unnormal Form* adalah model database yang paling sederhana. *Database* dengan model UNF akan mengalami masalah seperti redudansi data sehingga tidak memiliki efisiensi normalisasi *database*. Dengan kata lain, *database* dengan model UNF masih memiliki data yang berulang atau repeating group. Sehingga database perlu dilakukan normalisasi lebih lanjut kedalam bentuk *first normal form*.

#### 3. First Normal Form (1NF)

*First Normal Form* adalah relasi dimana perpotongan tiap baris dan kolom hanya berisi 1 nilai.

1NF memiliki aturan, yaitu:

- Relasi tidak boleh mengandung nilai ganda pada suatu cell dan tidak boleh ada *nested relation*.
- Tiap nama kolom harus *unique*.
- Nilai pada tiap baris harus *unique*.

- Tidak boleh ada *repeating groups*.

Yang perlu dilakukan pada tahap 1NF adalah:

- Bentuk relasi baru untuk setiap atribut multi-nilai atau *nested relation*.
- Hapus *repeating groups/information* ke dalam relasi terpisah dan mewakilinya sebagai baris. (Pisahkan nilai yang tidak tunggal ke baris baru jadi setiap baris dan kolom hanya boleh memiliki 1 nilai)

#### 4. Second Normal Form (2NF)

Langkah selanjutnya untuk normalisasi setelah 1NF adalah merubah *database* kedalam bentuk 2NF.

2NF memiliki aturan, yaitu:

- Relasi harus sudah dalam bentuk 1NF
- Semua atribut yang *non-primary key* harus *full functionally dependent* pada atribut *primary key*

Yang perlu dilakukan pada tahap 2NF adalah:

- Cari *primary key* pada relasi 1NF
- Identifikasi *functional dependency* pada relasi 1NF
- Jika terdapat *partial dependent*, dekomposisi dan buat relasi baru untuk setiap kunci parsial dengan atribut dependen-nya. (Menghilangkan *partial dependency*)
- Pastikan jaga relasi antara *primary key* yang asli dengan atribut apapun yang *functionally dependent* pada *primary key*.

#### 5. Third Normal Form (3NF)

Langkah selanjutnya untuk normalisasi setelah 2NF adalah merubah *database* kedalam bentuk 3NF.

3NF memiliki aturan, yaitu:

- Relasi harus sudah dalam bentuk 2NF
- Semua atribut *non-primary key* dari seluruh table tidak boleh tergantung kepada atribut lain yang merupakan *primary key*. (Tidak boleh ada *transitive dependency*)

Yang perlu dilakukan pada tahap 3NF adalah:

- Tentukan *primary key* pada relasi 2NF
- Identifikasi *functional dependency* pada relasi 2NF
- Jika terdapat *transitive dependent* pada *primary key*, dekomposisi dan buat relasi baru yang termasuk atribut *non-primary key* yang secara fungsional menentukan atribut *non-primary key* lainnya. (Menghilangkan *transitive dependency*)

## 6. Boyce-Codd Normal Form (BCNF)

Langkah selanjutnya untuk normalisasi setelah 3NF adalah merubah database kedalam bentuk BCNF.

BCNF memiliki aturan, yaitu:

- Relasi harus sudah dalam bentuk 3NF
- Untuk setiap *functional dependency*  $A \rightarrow B$ , maka A harus menjadi *super key*.

## 7. Entity Relationship Diagram(ERD)

ERD merupakan tahap terakhir yang kami lakukan pada project ini. ERD adalah jenis diagram struktural yang digunakan dalam desain database. ERD berisi berbagai simbol, notasi, bagan serta konektor yang memvisualisasikan dua informasi penting yaitu entitas utama dalam ruang lingkup sistem dan hubungan antar entitas.



## BAB III

### METODE PENELITIAN

#### A. Tools

Dalam menganalisis database, kita menggunakan tools sebagai berikut :

- Visual Paradigm
- Microsoft Excel
- Google Docs

#### B. Material

Variabel-variabel atau atribut-atribut yang ada dalam dataset kami adalah sebagai berikut:

1. *OrderID* yang menunjukan ID setiap transaksi.
2. *OrderDate* yang menunjukkan tanggal transaksi dilakukan.
3. *CustomerName* yang menunjukkan nama dari customer atau pembeli.
4. *ItemName* yang menunjukkan nama item yang dibeli.
5. *Quantity* yang menunjukkan kuantitas atau jumlah per item yang dibeli.
6. *ItemPrice* yang menunjukkan harga satuan tiap item.
7. *SubTotal* yang menunjukkan sub total biaya (didapatkan dari hasil perkalian antara *Quantity* dan *ItemPrice*).
8. *ItemTotal* yang menunjukkan harga total biaya dari pembelian item saja, didapatkan dari penjumlahan *SubTotal* setiap pembelian (belum termasuk biaya tambahan lainnya).
9. *Distance* yang menunjukkan jarak dari lokasi toko ke alamat pembeli.
10. *DeliveryFee* yang menunjukkan biaya pengiriman item dari lokasi toko ke alamat pembeli.
11. *ServiceAndOtherFee* yang menunjukkan biaya lain-lainnya seperti pajak, biaya pelayanan, dan lain sebagainya.

12. *GrandTotal* yang menunjukkan keseluruhan biaya yang harus dibayar oleh pembeli (didapatkan dari penjumlahan *ItemTotal*, *DeliveryFee*, dan *ServiceAndOtherFee*).
13. *DriverName* yang menunjukkan nama pengemudi yang bertanggung jawab mengantarkan item ke alamat pembeli.
14. *DriverLicencePlate* yang menunjukkan plat nomor kendaraan pengemudi.
15. *DriverVehicleType* yang menunjukkan jenis kendaraan dari pengemudi.
16. *StoreName* yang menunjukkan nama dari toko.
17. *StoreAddress* yang menunjukkan lokasi atau alamat dari toko.
18. *StoreUrbanVillage* yang menunjukkan kelurahan dari toko.
19. *StoreSubDistrict* yang menunjukkan kecamatan dari toko.
20. *StoreDistrict* yang menunjukkan kabupaten atau kota dari toko.
21. *StoreProvince* yang menunjukkan provinsi dari toko.
22. *StorePostalCode* yang menunjukkan kode pos dari toko.
23. *CustomerAddress* yang menunjukkan lokasi atau alamat dari pembeli.
24. *CustomerUrbanVillage* yang menunjukkan kelurahan dari pembeli.
25. *CustomerSubDistrict* yang menunjukkan kecamatan dari pembeli.
26. *CustomerDistrict* yang menunjukkan kabupaten dari pembeli.
27. *CustomerProvince* yang menunjukkan provinsi dari pembeli.
28. *CustomerPostalCode* yang menunjukkan kode pos dari pembeli.
29. *PaymentType* yang menunjukkan tipe pembayaran yang dilakukan oleh pembeli.
30. *DeliveredAt* yang menunjukkan waktu pengiriman yang dilakukan oleh pengemudi (waktu pengemudi bergerak ke alamat pembeli).
31. *ReceivedAt* yang menunjukkan waktu pembeli menerima pesanan.

Data dari struk memiliki berbagai atribut serta informasi guna dianalisa relasi antar entitasnya sehingga dapat terbentuk suatu diagram ERD.

### C. Implementasi Metode

Metode yang akan kami gunakan dalam database adalah sebagai berikut:

- Analisis yang dibutuhkan:

Menganalisa struk yang ada untuk mencatat / merecord transaksi pembelian.

- Desain Database:

Peneliti akan merancang *database* yang cocok untuk struk yang peneliti punya. Kemudian, peneliti akan menyusun informasi-informasi/data yang ada dalam struk ke dalam bentuk tabel untuk setiap entitas dan menormalisasikan tabel-tabel data tersebut. Selain itu, data juga akan ditransformasikan menjadi ERD yang mampu menampilkan relasi dan properti dari masing-masing entitas.

#### **D. Prosedur Penelitian**

Proses penelitian saat mengerjakan database adalah sebagai berikut:

- Menganalisis dan mengamati struk.
- Membuat tabel dan mengisinya dengan data-data yang terdapat pada struk.
- Mengimplementasikan normalisasi pada database asli mengikuti aturan-aturan dari *Unnormalized Form* (UNF) hingga *Boyce-Codd Normalization Form* (BCNF).
- Menggambarkan koneksi antar entitas pada basis data melalui *Entity Relationship Diagram* (ERD).

#### **E. Metode Mengumpulkan Data**

##### **1. Observasi**

Metode observasi adalah metode yang digunakan untuk pengumpulan data dengan cara melakukan pengambilan data melalui pengamatan langsung terhadap situasi. Kami melakukan observasi non participant yang artinya kami tidak ikut secara langsung dalam kegiatan dan proses yang sedang diamati.

##### **2. Metode Studi Dokumen**

Metode ini adalah metode pengumpulan data dengan cara mengkaji dan menganalisis berbagai jenis dokumen. metode ini digunakan untuk mencari dan menganalisis berbagai masalah dan pemahaman teoritis untuk menemukan penjelasan pertanyaan penelitian.

## BAB IV

### IMPLEMENTASI

Primary Key	
Foreign Key	
Composite Key	

#### UNF (Unnormalized Form)

Pada tahap ini, data-data yang tertera pada struk diubah bentuknya menjadi tabel. Setiap item yang dibeli oleh *customer* akan dimasukkan ke masing-masing baris tabel. Hal ini dilakukan karena pada struk tertulis bahwa jumlah item yang dibeli oleh *customer* hanya berjumlah 2 item. Apabila data setiap item yang dibeli ditulis pada suatu atributnya masing-masing (*ItemName*, *Item2Name*), maka probabilitas terdapat anomali pada tabel akan semakin besar. Selain itu, apabila bertemu dengan kondisi dimana *customer* membeli lebih dari 2 item atau kurang dari 2 item, maka kondisi yang terjadi adalah bisa saja salah satu *field* menjadi kosong/*null* atau kekurangan kolom untuk menyimpan nama-nama item yang dibeli. Misalkan berdasarkan struk yang ada, di-*create* 2 kolom (*Item1Name*, *Item2Name*) yang berisikan data-data item yang dibeli pada struk tersebut. Lalu, ditemukan kondisi dimana seorang *customer* hanya membeli 1 item, maka salah satu *field* (*Item2Name*) akan berisikan *null value*. Lalu, apabila ditemukan dengan kondisi dimana seorang *customer* membeli lebih dari 2 item, kemungkinan yang dapat terjadi adalah akan di-*create* kolom-kolom baru untuk menampung nama-nama dari item yang dibeli oleh *customer*. Kemungkinan lainnya adalah meng-*create* baris-baris data yang baru untuk menampung nama-nama item atau dalam suatu baris data dituliskan nama item dalam sebuah kolom yang sama dengan *delimiter*, contoh Trio Hotdakk Mozzarella;Pure Cocoa, yang dimana penulisan data tersebut tidak diperbolehkan. Apabila terus menerus meng-*create* kolom yang baru untuk menyimpan nama-nama dari data, maka *time complexity* akan terus bertambah, memperbesar kemungkinan terdapat adanya *missing value* dan data anomali, dan tidak mengecilkan kemungkinan terdapat adanya redundansi data. Oleh karena itu, pada tahap ini, dibentuklah tabel berbentuk UNF sebagai berikut:

#### - Transaction Table

OrderID	OrderDate	CustomerName	ItemID	ItemName	Quantity	ItemPrice
F-1942805643	08-12-2022	Fepri Putra	IE94567218300	Trio Hotdakk Mozarella	1	67000
F-1942805643	08-12-2022	Fepri Putra	IE84167924386	Pure Cocoa	1	26000

SubTotal	ItemTotal	Distance	DeliveryFee	ServiceAndOtherFee	GrandTotal
67000	93000	3.4	13000	9000	115000
26000	93000	3.4	13000	9000	115000

DriverName	DriverLicensePlate	DriverVehicleType	StoreName	StoreAddress	StoreUrbanVillage
Agam Maulana	B6394WUF	Honda Beat	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jl. Pamulang Permai 1	Pamulang Barat
Agam Maulana	B6394WUF	Honda Beat	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jl. Pamulang Permai 1	Pamulang Barat

StoreSubDistrict	StoreDistrict	StoreProvince	StorePostalCode	CustomerAddress	CustomerUrbanVillage
Pamulang	Tangerang Selatan	Banten	15417	Jl. Mawar 3 Blok C3 No. 1	Kedaung
Pamulang	Tangerang Selatan	Banten	15417	Jl. Mawar 3 Blok C3 No. 1	Kedaung

Customer	Customer	Customer	Customer	Payment	DeliveredAt
----------	----------	----------	----------	---------	-------------

SubDistrict	District	Province	PostalCode	Type	
Pamulang	Tangerang Selatan	Banten	15415	Gopay	17:43
Pamulang	Tangerang Selatan	Banten	15415	Gopay	17:43

ReceivedAt	DeliveryTime
18:06	23
18:06	23

#### Functional Dependency :

**OrderID** → OrderDate, CustomerName, Quantity, ItemTotal, Distance, DeliveryFee, ServiceAndOtherFee, GrandTotal, DriverName, DriverLicensePlate, DriverVehicleType, StoreName, StoreAddress, StoreUrbanVillage, StoreSubDistrict, StoreDistrict, StoreProvince, StorePostalCode, CustomerAddress, CustomerUrbanVillage, CustomerSubDistrict, CustomerDistrict, CustomerProvince, CustomerPostalCode, PaymentType, DeliveredAt, ReceivedAt, DeliveryTime

**OrderID, ItemID** → ItemName, ItemPrice, Quantity

#### 1NF (1st Normal Form)

Pada tahap 1NF ini, yang dilakukan adalah menghilangkan data yang bersifat redundan serta bersifat perhitungan atau kalkulasi, dan menentukan *Primary Key* (PK) dan *Foreign Key* (FK) pada masing-masing tabel. Dilakukan juga penambahan kolom-kolom pada tabel. Untuk menghilangkan data yang bersifat redundan, tabel *Transaction* dipecah menjadi 2, yaitu tabel atau *TransactionHeader* dan tabel *TransactionDetail*, serta ditentukan *primary key* dan *foreign key* pada masing-masing tabel. Selanjutnya, untuk menghilangkan data yang bersifat perhitungan, dihilangkan kolom *SubTotal* (diperoleh dari perkalian *ItemPrice* dan *Quantity*), kolom *ItemTotal* (diperoleh dari jumlah *SubTotal* setiap transaksi), kolom *GrandTotal* (diperoleh dari penjumlahan *ItemTotal*, *DeliveryFee*, dan *ServiceAndOtherFee*), dan kolom *DeliveryTime* (diperoleh dengan dari pengurangan *DeliveredAt* dengan *ReceivedAt*). Hasil yang didapatkan setelah melakukan langkah-langkah yang sudah disebutkan adalah sebagai berikut:

#### - TransactionHeader Table

TransactionHeader				
OrderID	OrderDate	CustomerID	CustomerName	DriverID
F-1942805643	08-12-2022	CU6182349271	Fepri Putra	DR1234854718

TransactionHeader				
DriverName	Driver LicensePlate	DriverVehicleType	StoreID	StoreName
Agam Maulana	B6394WUF	Honda Beat	ST8945173962	Chatime X Cupbop, LP Pamulang

TransactionHeader				
StoreAddress	Store UrbanVillageID	StoreUrbanVillage	Store SubDistrictID	Store SubDistrict
Komplek Pamulang Permai, Jl. Pamulang Permai 1	KL001	Pamulang Barat	KC001	Pamulang

TransactionHeader				
StoreDistrictID	StoreDistrict	Store ProvinceID	StoreProvince	Store PostalCodeID
KK001	Tangerang Selatan	PV001	Banten	PC001

TransactionHeader				
StorePostalCode	ServiceAndOther Fee	Customer Address	Customer UrbanVillageID	Customer UrbanVillage
15417	9000	Jl. Mawar 3 Blok C3 No. 1	KL002	Kedaung

TransactionHeader				
Customer SubDistrictID	Customer SubDistrict	CustomerDistrictID	CustomerDistrict	Customer ProvinceID

KC001	Pamulang	KK001	Tangerang Selatan	PV001
-------	----------	-------	-------------------	-------

TransactionHeader				
CustomerProvince	CustomerPostalCodeID	CustomerPostalCode	Distance	PaymentType
Banten	PC002	15415	3.4	Gopay

TransactionHeader		
DeliveryFee	DeliveredAt	ReceivedAt
13000	17:43	18:06

#### Functional Dependency:

**OrderID** → OrderDate, CustomerID, CustomerName, DriverID, DriverName, DriverLicensePlate, DriverVehicleType, StoreID, StoreName, StoreAddress, StoreUrbanVillageID, StoreSubDistrictID, StoreSubDistrict, StoreDistrictID, StoreDistrict, StoreProvinceID, StoreProvince, StorePostalCodeID, StorePostalCode, ServiceAndOtherFee, CustomerAddress, CustomerUrbanVillageID, CustomerUrbanVillage, CustomerSubDistrictID, CustomerSubDistrict, CustomerDistrictID, CustomerDistrict, CustomerProvinceID, CustomerProvince, CustomerCodeID, Distance, PaymentType, DeliveryFee, DeliveredAt, ReceivedAt

#### - TransactionDetail Table

TransactionDetail				
OrderID	ItemID	ItemName	ItemPrice	Quantity
F-1942805643	IE94567218300	Trio Hotdakk Mozarella	67000	1
F-1942805643	IE84167924386	Pure Cocoa	26000	1

#### Functional Dependency:

**OrderID, ItemID** → ItemName, ItemPrice, Quantity

**ItemID** → ItemName, ItemPrice

**OrderID** → Quantity



## 2NF (2nd Normal Form)

Tabel sebelumnya sudah memenuhi 1NF akan tetapi belum memenuhi 2NF. Pada tabel tersebut masih terjadi *violation* pada aturan 2NF dikarenakan terdapat atribut non *primary key* yang bergantung pada subset kunci kandidat (misalnya pada *ItemName* dan *ItemPrice* yang bergantung pada *ItemID*). *ItemName* dan *ItemPrice* berada di kondisi di mana kedua atribut bukan merupakan *primary key*, namun keberadaannya bergantung pada *ItemID* yang di mana *ItemID* sendiri merupakan *composite key* atau termasuk sebagai salah satu *primary key* pada tabel *TransactionDetail* yang terbentuk pada *First Normal Form* (1NF). Oleh karena itu, agar data dapat memenuhi aturan 2NF, tabel akan dipisahkan menjadi tiga tabel yakni tabel *TransactionHeader*, tabel *TransactionDetail*, dan tabel *MsItem*. Tabel *TransactionDetail* yang sebelumnya memiliki *composite key* (*OrderID* dan *ItemID*) akan dipisahkan menjadi dua tabel. Tabel yang pertama adalah tabel *TransactionDetail* yang memiliki atribut *OrderID* dan *ItemID* sebagai *primary key* dan atribut-atribut tersebut dapat menentukan *Quantity*. Tabel kedua adalah tabel *MsItem* yang berisi atribut *ItemID* sebagai *primary key* dan atribut tersebut dapat menentukan atribut lainnya yaitu *ItemName* dan *ItemPrice*. Rincian dari tabel-tabel tersebut adalah sebagai berikut :

### - TransactionHeader Table

TransactionHeader				
OrderID	OrderDate	CustomerID	CustomerName	DriverID
F-1942805643	08-12-2022	CU6182349271	Fepri Putra	DR1234854718

TransactionHeader				
DriverName	Driver LicensePlate	DriverVehicleType	StoreID	StoreName
Agam Maulana	B6394WUF	Honda Beat	ST8945173962	Chatime X Cupbop, LP Pamulang

TransactionHeader				
StoreAddress	StoreUrban	StoreUrbanVillage	Store	Store

	VillageID		SubDistrictID	SubDistrict
Komplek Pamulang Permai, Jl. Pamulang Permai 1	KL001	Pamulang Barat	KC001	Pamulang

TransactionHeader				
StoreDistrictID	StoreDistrict	StoreProvinceID	StoreProvince	StorePostalCodeID
KK001	Tangerang Selatan	PV001	Banten	PC001

TransactionHeader				
StorePostalCode	ServiceAndOtherFee	CustomerAddress	CustomerUrbanVillageID	CustomerUrbanVillage
15417	9000	Jl. Mawar 3 Blok C3 No. 1	KL002	Kedaung

TransactionHeader				
CustomerSubDistrictID	CustomerSubDistrict	CustomerDistrictID	CustomerDistrict	CustomerProvinceID
KC001	Pamulang	KK001	Tangerang Selatan	PV001

TransactionHeader				
CustomerProvince	CustomerPostalCodeID	CustomerPostalCode	Distance	PaymentType
Banten	PC002	15415	3.4	Gopay

TransactionHeader		
DeliveryFee	DeliveredAt	ReceivedAt
13000	17:43	18:06

**Functional Dependency:**

**OrderID** → OrderDate, CustomerID, DriverID, StoreID, Distance, PaymentType, DeliveryFee, DeliveredAt, ReceivedAt

**CustomerID** → CustomerName, CustomerAddress, CustomerUrbanVillageID, CustomerUrbanVillage, CustomerSubDistrictID, CustomerSubDistrict, CustomerDistrictID, CustomerDistrict, CustomerProvinceID, CustomerProvince, CustomerCodeID

**DriverID** → DriverName, DriverLicensePlate, DriverVehicleType

**StoreID** → StoreName, StoreAddress, StoreUrbanVillageID, StoreSubDistrictID, StoreSubDistrict, StoreDistrictID, StoreDistrict, StoreProvinceID, StoreProvince, StorePostalCodeID, StorePostalCode, ServiceAndOtherFee

- **TransactionDetail Table**

<i>TransactionDetail</i>		
OrderID	ItemID	Quantity
F-1942805643	IE94567218300	1
F-1942805643	IE84167924386	1

**Functional Dependency:**

**OrderID, ItemID** → Quantity

- **MsItem Table**

<i>MsItem</i>		
ItemID	ItemName	ItemPrice
IE94567218300	Trio Hotdakk Mozarella	67000
IE84167924386	Pure Cocoa	26000

**Functional Dependency:**

**ItemID** → ItemName, ItemPrice

**3NF (3rd Normal Form)**

Dalam *Third Normal Form* (3NF), aturan utama yang harus dipenuhi ialah bahwa tabel harus memenuhi 2NF dan tidak ada *transitive functional dependency*. Selama tabel masih dalam keadaan 2NF maka masih memungkinkan untuk terjadinya *update* anomali. Secara sederhana *transitive functional dependency* terjadi ketika terdapat ketergantungan antara *non-prime* atribut dengan *non-prime* atribut melalui atribut *non-prime* lainnya. Pada tabel sebelumnya terdapat *transitive functional dependency* pada tabel *TransactionHeader*. Secara fungsional, atribut *OrderID* memiliki ketergantungan pada semua atribut lainnya, dalam hal ini *OrderID*

berperan sebagai *primary key*, akan tetapi, terdapat ketergantungan antara *non prime* atribut dengan *non prime* attribute misalnya, *DriverName*, *DriverLicensePlate*, dan *DriverVehicleType* bergantung pada atribut *DriverID* padahal *DriverID* tidak berperan sebagai atribut *primary*. Kemudian atribut *non-prime* seperti *StoreName*, *StoreAddress*, *StoreUrbanVillageID*, *StoreUrbanVillage*, *StoreSubDistrictID*, *StoreSubDistrict*, *StoreDistrictID*, *StoreDistrict*, *StoreProvinceID*, *StoreProvince*, *StorePostalCodeID*, *StorePostalCode*, dan *ServiceAndOtherFee* memiliki ketergantungan pada *non-prime* attribute yakni *StoreID*. Selanjutnya terdapat lagi atribut *non prime* seperti *CustomerName*, *CustomerAddress*, *CustomerUrbanVillageID*, *CustomerUrbanVillage*, *CustomerSubDistrictID*, *CustomerSubDistrict*, *CustomerDistrictID*, *CustomerDistrict*, *CustomerProvinceID*, *CustomerProvince*, *CustomerPostalCodeID*, dan *CustomerPostalCode* yang bergantung pada atribut *non prime* yaitu *CustomerID*. Oleh karena terdapat ketergantungan antara atribut *non prime* dengan atribut *non prime*, perlu dibentuk entitas baru bernama *MsDriver*, *MsStore*, dan *MsCustomer*. Atribut *DriverID* akan menjadi *primary key* untuk tabel *MsDriver*, atribut *StoreID* akan menjadi *primary key* tabel *MsStore*, atribut *CustomerID* akan menjadi *primary key* tabel *MsCustomer*.

#### - TransactionHeader Table

TransactionHeader				
OrderID	CustomerID	DriverID	StoreID	Distance
F-1942805643	CU6182349271	DR1234854718	ST8945173962	3.4

TransactionHeader				
DeliveryFee	PaymentType	DeliveryAt	ReceivedAt	OrderDate
13000	Gopay	17:43	18:06	08-12-2022

#### Functional Dependency:

**OrderID** → CustomerID, DriverID, StoreID, Distance, DeliveryFee, PaymentType, DeliveredAt, ReceivedAt, OrderDate

#### - TransactionDetail Table

TransactionDetail		
OrderID	ItemID	Quantity

F-1942805643	IE94567218300	1
F-1942805643	IE84167924386	1

### Functional Dependency:

**OrderID, ItemID** → Quantity

#### - MsItem Table

MsItem		
ItemID	ItemName	ItemPrice
IE94567218300	Trio Hotdakk Mozarella	67000
IE84167924386	Pure Cocoa	26000

### Functional Dependency:

**ItemID** → ItemName, ItemPrice

#### - MsDriverTable

MsDriver			
DriverID	DriverName	DriverLicensePlate	DriverVehicleType
DR1234854718	Agam Maulana	B6394WUF	Honda Beat

### Functional Dependency:

**DriverID** → DriverName, DriverLicensePlate, DriverVehicleType

#### - MsStoreTable

MsStore					
StoreID	StoreName	StoreAddress	Store UrbanVillageID	Store UrbanVillage	
ST8945173962	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jl. Pamulang Permai 1	KL001	Pamulang Barat	

MsStore				
StoreSubDistrictID	StoreSubDistrict	StoreDistrictID	StoreDistrict	StoreProvinceID
KC001	Pamulang	KK001	Tangerang Selatan	PV001

MsStore			
StoreProvince	StorePostalCodeID	StorePostalCode	ServiceAndOtherFee
Banten	PC001	15417	9000

#### Functional Dependency:

**StoreID** → StoreName, StoreAddress, StoreUrbanVillageID, StoreUrbanVillage, StoreSubDistrictID, StoreSubDistrict, StoreDistrictID, StoreDistrict, StoreProvinceID, StoreProvince, StorePostalCodeID, StorePostalCode, ServiceAndOtherFee

**StoreUrbanVillageID** → StoreUrbanVillage

**StoreSubDistrictID** → StoreSubDistrict

**StoreDistrictID** → StoreDistrict

**StoreProvinceID** → StoreProvince

**StorePostalCodeID** → StorePostalCode

#### - MsCustomerTable

MsCustomer				
CustomerID	CustomerName	CustomerAddress	CustomerUrbanVillageID	CustomerUrbanVillage
CU6182349271	Fepri Putra	Jl. Mawar 3 Blok C3 No. 1	KL002	Kedaung

MsCustomer
------------

Customer SubDistrictID	Customer SubDistrict	Customer DistrictID	Customer District	Customer ProvinceID
KC001	Pamulang	KK001	Tangerang Selatan	PV001

MsCustomer		
Customer Province	Customer PostalCodeID	Customer PostalCode
Banten	PC002	15415

### Functional Dependency:

**CustomerID** → CustomerName, CustomerAddress, CustomerUrbanVillageID, CustomerUrbanVillage, CustomerSubDistrictID, CustomerSubDistrict, CustomerDistrictID, CustomerDistrict, CustomerProvinceID, CustomerProvince, CustomerPostalCodeID, CustomerPostalCode

**CustomerUrbanVillageID** → CustomerUrbanVillage

**CustomerSubDistrictID** → CustomerSubDistrict

**CustomerDistrictID** → CustomerDistrict

**CustomerProvinceID** → CustomerProvince

**CustomerPostalCodeID** → CustomerPostalCode

### BCNF (Boyce-Codd Normal Form)

*Boyce-Codd Normal Form* (BCNF) merupakan kelanjutan dari 3NF atau sering juga disebut dengan 3.5NF. BCNF bersifat lebih tegas dibandingkan 3NF. Aturan pada BCNF ini adalah bahwa tabel harus memenuhi aturan 3NF serta untuk setiap functional dependency ( $X \rightarrow Y$ ), maka X harus merupakan *super key*. Pada tabel sebelumnya, masih terdapat functional dependency ( $X \rightarrow Y$ ) dimana X belum merupakan *super key*. Seperti halnya pada functional dependency:  $CustomerUrbanVillageID \rightarrow CustomerUrbanVillage$ ,  $CustomerSubDistrictID \rightarrow CustomerSubDistrict$ ,  $CustomerDistrictID \rightarrow CustomerDistrict$ ,  $CustomerProvinceID \rightarrow CustomerProvince$ ,  $CustomerPostalCodeID \rightarrow CustomerPostalCode$ , X belum merupakan *super key* sehingga dibentuk tabel baru yaitu *MsUrbanVillage*, *MsSubDistrict*, *MsDistrict*, *MsProvince*, serta *MsPostalCode* sehingga aturan BCNF dapat terpenuhi.

- TransactionHeader Table

TransactionHeader				
OrderID	CustomerID	DriverID	StoreID	Distance
F-1942805643	CU6182349271	DR1234854718	ST8945173962	3.4

TransactionHeader				
DeliveryFee	PaymentType	DeliveredAt	ReceivedAt	OrderDate
13000	Gopay	17:43	18:06	08-12-2022

**Functional Dependency:**

**OrderID** → CustomerID, DriverID, StoreID, Distance, DeliveryFee, PaymentType, DeliveredAt, ReceivedAt, OrderDate

- TransactionDetail Table

TransactionDetail		
OrderID	ItemID	Quantity
F-1942805643	IE94567218300	1
F-1942805643	IE84167924386	1

**Functional Dependency:**

**OrderID** → ItemID, Quantity

- MsItem Table

MsItem		
ItemID	ItemName	ItemPrice
IE94567218300	Trio Hotdakk Mozzarella	67000
IE84167924386	Pure Cocoa	26000

**Functional Dependency:**

**ItemID** → ItemName, ItemPrice

- MsDriver Table

MsDriver
----------



DriverID	DriverName	DriverLicensePlate	DriverVehicleType
DR1234854718	Agam Maulana	B6394WUF	Honda Beat

#### Functional Dependency:

**DriverID** → DriverName, DriverLicensePlate, DriverVehicleType

#### - MsStore Table

MsStore				
StoreID	StoreName	StoreAddress	StoreUrbanVillageID	StoreSubDistrictID
ST8945173962	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jl. Pamulang Permai 1	KL001	KC001

MsStore			
StoreDistrictID	StoreProvinceID (FK)	StorePostalCodeID	ServiceAndOtherFee
KK001	PV001	PC001	9000

#### Functional Dependency:

**StoreID** → StoreName, StoreAddress, StoreUrbanVillageID, StoreSubDistrictID, StoreDistrictID, StoreProvinceID, StorePostalCodeID, ServiceAndOtherFee

#### - MsCustomer Table

MsCustomer				
CustomerID	Customer Name	Customer Address	Customer UrbanVillageID	Customer SubDistrictID
CU6182349271	Fepri Putra	Jl. Mawar 3 Blok C3 No. 1	KL002	KC001

MsCustomer		
CustomerDistrictID	CustomerProvinceID	CustomerPostalCodeID
KK001	PV001	PC002

**Functional Dependency:**

**CustomerID** → CustomerName, CustomerAddress, CustomerUrbanVillage, CustomerSubDistrictID, CustomerDistrictID, CustomerProvinceID, CustomerPostalCodeID

- **MsUrbanVillage Table**

<i>MsUrbanVillage</i>	
UrbanVillageID	UrbanVillageName
KL001	Pamulang Barat
KL002	Kedaung

**Functional Dependency:**

**UrbanVillageID** → UrbanVillageName

- **MsSubDistrict Table**

<i>MsSubDistrict</i>	
SubDistrictID	SubDistrictName
KC001	Pamulang

**Functional Dependency:**

**SubDistrictID** → SubDistrictName

- **MsDistrict Table**

<i>MsDistrict</i>	
DistrictID	DistrictName
KK001	Tangerang Selatan

**Functional Dependency:**

**DistrictID** → DistrictName

- MsProvince Table

<i>MsProvince</i>	
ProvinceID	ProvinceName
PV001	Banten

**Functional Dependency:**

**ProvinceID** → ProvinceName

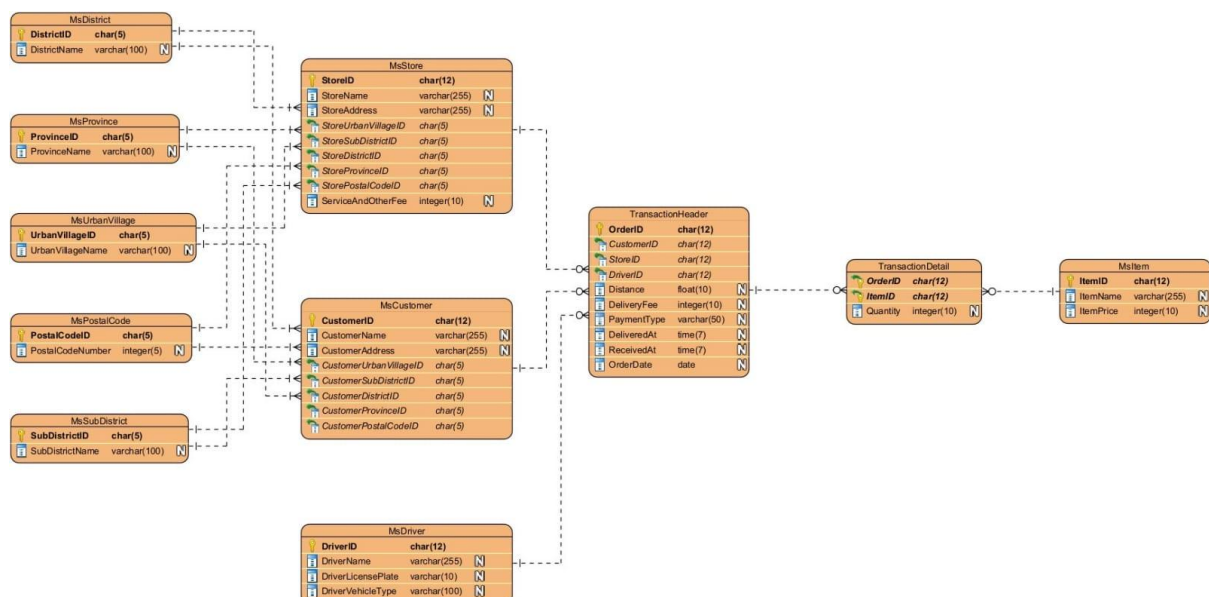
- MsPostalCode Table

<i>MsPostalCode</i>	
PostalCodeID	PostalCodeNumber
PC001	15417
PC002	15415

**Functional Dependency:**

**PostalCodeID** → PostalCodeNumber

**Entity Relationship Diagram (ERD) in Crow's Foot Notation**



Link Gambar ERD:

<https://drive.google.com/file/d/1F6veBBpWwDclYUN0iDWO7PGW1plrha5o/view?usp=sharing>

### **Source Code**

```
CREATE DATABASE gojek_food

CREATE TABLE MsDistrict(
    DistrictID CHAR(5) PRIMARY KEY NOT NULL,
    DistrictName VARCHAR(100) NOT NULL
);

CREATE TABLE MsProvince(
    ProvinceID CHAR(5) PRIMARY KEY NOT NULL,
    ProvinceName VARCHAR(100) NOT NULL
);

CREATE TABLE MsUrbanVillage(
    UrbanVillageID CHAR(5) PRIMARY KEY NOT NULL,
    UrbanVillageName VARCHAR(100) NOT NULL
);

CREATE TABLE MsPostalCode(
    PostalCodeID CHAR(5) PRIMARY KEY NOT NULL,
    PostalCodeNumber INT(5) NOT NULL
);

CREATE TABLE MsSubDistrict(
    SubDistrictID CHAR(5) PRIMARY KEY NOT NULL,
    SubDistrictName VARCHAR(100) NOT NULL
);

INSERT INTO MsDistrict VALUES
    ("KK001", "Tangerang Selatan"),
    ("KK002", "Jakarta Selatan"),
    ("KK003", "Jakarta Timur"),
    ("KK004", "Jakarta Pusat"),
    ("KK005", "Jakarta Barat"),
    ("KK006", "Cibadak"),
    ("KK007", "Bandung"),
    ("KK008", "Bekasi"),
```

```
("KK009", "Bogor"),  
("KK010", "Karawang");
```

```
INSERT INTO MsProvince VALUES  
("PV001", "Banten"),  
("PV002", "DKI Jakarta"),  
("PV003", "Jawa Barat"),  
("PV004", "Jawa Tengah"),  
("PV005", "Daerah Istimewa Yogyakarta"),  
("PV006", "Jawa Timur"),  
("PV007", "Bali"),  
("PV008", "Nusa Tenggara Timur"),  
("PV009", "Nusa Tenggara Barat"),  
("PV010", "Kepulauan Riau");
```

```
INSERT INTO MsUrbanVillage VALUES  
("KL001", "Pamulang Barat"),  
("KL002", "Kedaung"),  
("KL003", "Cipete Selatan"),  
("KL004", "Duren Sawit"),  
("KL005", "Cikini"),  
("KL006", "Slipi"),  
("KL007", "Margasari"),  
("KL008", "Kelapa Indah"),  
("KL009", "Periuk Jaya"),  
("KL010", "Kebon Besar");
```

```
INSERT INTO MsPostalCode VALUES  
("PC001", 15417),  
("PC002", 15415),  
("PC003", 15418),  
("PC004", 15419),  
("PC005", 15420),  
("PC006", 15421),  
("PC007", 15422),  
("PC008", 62801),  
("PC009", 26375),  
("PC010", 46257);
```

```
INSERT INTO MsSubDistrict VALUES  
("KC001", "Pamulang"),  
("KC002", "Cilandak"),  
("KC003", "Duren Sawit"),  
("KC004", "Menteng"),
```

```

("KC005", "Palmerah"),
("KC006", "Buahbatu"),
("KC007", "Batucapeper"),
("KC008", "Tangerang"),
("KC009", "Periuk"),
("KC010", "Palmerah");

```

```

CREATE TABLE MsStore(
    StoreID CHAR(12) PRIMARY KEY NOT NULL,
    StoreName VARCHAR(255) NOT NULL,
    StoreAddress VARCHAR(255) NOT NULL,
    StoreUrbanVillageID CHAR(5) NOT NULL,
    StoreSubDistrictID CHAR(5) NOT NULL,
    StoreDistrictID CHAR(5) NOT NULL,
    StoreProvinceID CHAR(5) NOT NULL,
    StorePostalCodeID CHAR(5) NOT NULL,
    ServiceAndOtherFee INT(10),

    FOREIGN KEY (StoreUrbanVillageID) REFERENCES
msurbanvillage(UrbanVillageID),
    FOREIGN KEY (StoreSubDistrictID) REFERENCES
mssubdistrict(SubDistrictID),
    FOREIGN KEY (StoreDistrictID) REFERENCES
msdistrict(DistrictID),
    FOREIGN KEY (StoreProvinceID) REFERENCES
msprovince(ProvinceID),
    FOREIGN KEY (StorePostalCodeID) REFERENCES
mspostalcode(PostalCodeID)
);

```

```

CREATE TABLE MsCustomer(
    CustomerID CHAR(12) PRIMARY KEY NOT NULL,
    CustomerName VARCHAR(255) NOT NULL,
    CustomerAddress VARCHAR(255) NOT NULL,
    CustomerUrbanVillageID CHAR(5) NOT NULL,
    CustomerSubDistrictID CHAR(5) NOT NULL,
    CustomerDistrictID CHAR(5) NOT NULL,
    CustomerProvinceID CHAR(5) NOT NULL,
    CustomerPostalCodeID CHAR(5) NOT NULL,

    FOREIGN KEY (CustomerUrbanVillageID) REFERENCES
msurbanvillage(UrbanVillageID),
    FOREIGN KEY (CustomerSubDistrictID) REFERENCES
mssubdistrict(SubDistrictID),

```

```

        FOREIGN KEY (CustomerDistrictID) REFERENCES
msdistrict(DistrictID),
        FOREIGN KEY (CustomerProvinceID) REFERENCES
msprovince(ProvinceID),
        FOREIGN KEY (CustomerPostalCodeID) REFERENCES
mspostalcode(PostalCodeID)
);

CREATE TABLE MsDriver(
    DriverID CHAR(12) PRIMARY KEY NOT NULL,
    DriverName VARCHAR(255) NOT NULL,
    DriverLicensePlate VARCHAR(10) NOT NULL,
    DriverVehicleType VARCHAR(100) NOT NULL
);

INSERT INTO MsStore VALUES
    ("ST8945173962", "Chatime X Cupbop, LP Pamulang", "Komplek
Pamulang Permai, Jalan Pamulang Permai 1", "KL001", "KC001",
"KK001", "PV001", "PC001", 9000),
    ("ST6370240598", "Martabak Pecenongan 78", "Jalan Entong
Gendut Nomor 94", "KL003", "KC005", "KK003", "PV005", "PC002",
12000),
    ("ST3365348135", "Rujak & Asinan Buah Aa", "Jalan
Sulaiman, Gang Amal 4 Nomor 16", "KL002", "KC005", "KK010",
"PV003", "PC009", 9000),
    ("ST3300670796", "Istana Bubur Ketan", "Jalan Kebayoran
Lama Raya Nomor 7", "KL008", "KC009", "KK002", "PV009",
"PC007", 10000),
    ("ST2241853257", "Cold Brownies Caja", "Komplek Podomoro
City, Jl. Letjen S. Parman Kavling 28", "KL006", "KC010",
"KK005", "PV002", "PC003", 15000),
    ("ST6848506081", "Choipan Dan Kaloci PHo-PHo", "Jalan
Tanjung Duren Selatan Gang 8 Nomor 7", "KL009", "KC003",
"KK009", "PV006", "PC002", 10000),
    ("ST1197623553", "Fuko Fuko Cheese Tea", "Komplek Podomoro
City, Jl. Letjen S. Parman Kavling 28", "KL006", "KC010",
"KK005", "PV002", "PC003", 13000),
    ("ST4241021513", "Jakarta Cheese Factory", "Komplek
Podomoro City, Jl. Letjen S. Parman Kavling 28", "KL006",
"KC010", "KK005", "PV002", "PC003", 15000),
    ("ST7102815101", "Lekker Story", "Komplek Podomoro City,
Jl. Letjen S. Parman Kavling 28", "KL006", "KC010", "KK005",
"PV002", "PC003", 9000),

```

```
("ST7262235209", "Sweet Truth", "Komplek Podomoro City,  
Jl. Letjen S. Parman Kavling 28", "KL006", "KC010", "KK005",  
"PV002", "PC003", 12000);
```

```
INSERT INTO MsCustomer VALUES
```

```
("CU6182349271", "Fepri Putra", "Jl Mawar 3 Blok C3 No.  
1", "KL002", "KC001", "KK001", "PV001", "PC002"),  
("CU8200800827", "Ahmad Kurniawan", "Jl Melati 1 Blok A  
No. 10", "KL009", "KC010", "KK010", "PV003", "PC010"),  
("CU6018155509", "Budi Adi", "Jl Melati 1 Blok A No.10",  
"KL009", "KC010", "KK010", "PV003", "PC010"),  
("CU5839418311", "Puspita Ayu", "Jl Mawar 2 Blok A4 No.5",  
"KL002", "KC001", "KK001", "PV001", "PC002"),  
("CU3872338300", "Novita Dyah", "Jl Mawar 3 Blok C3 No.  
6", "KL002", "KC001", "KK001", "PV001", "PC002"),  
("CU6951981065", "Maria Pratiwi", "Jl Mawar 5 Blok D4 No.  
15", "KL002", "KC001", "KK001", "PV001", "PC002"),  
("CU9720788482", "Kartika Wulandari", "Jl Bugenville Blok  
D No.25", "KL007", "KC008", "KK008", "PV002", "PC005"),  
("CU6226426715", "I Ketut", "Jl Melati Kav. 97", "KL009",  
"KC010", "KK010", "PV003", "PC010"),  
("CU2529474198", "Nurul Putri", "Jl Tulip Blok B No. 1",  
"KL004", "KC005", "KK005", "PV003", "PC002"),  
("CU5361522657", "Arif Muhammad", "Jl Tulip Blok D No.15",  
"KL004", "KC005", "KK005", "PV003", "PC002");
```

```
INSERT INTO MsDriver VALUES
```

```
("DR1234854718", "Agam Maulana", "B6394WUF", "Honda  
Beat"),  
("DR3109658692", "Aditya Nugroho", "B4560PRG", "Suzuki  
Nex"),  
("DR5497501521", "Yunita Rina", "B6488AAV", "Honda Beat"),  
("DR6056883237", "Putri Rahayu", "B2208MLO", "Honda  
Beat"),  
("DR4116616002", "Rina Maya", "B5259JKJ", "Yamaha Byson"),  
("DR7006973912", "Arief Handayano", "B6009ACM", "Yamaha  
Mio S"),  
("DR7024063722", "Citra Munaroh", "B6086BQN", "Honda Revo  
Fit"),  
("DR9005452865", "Puji Devi", "B1608WAE", "Honda Beat"),  
("DR6841368364", "Arief Kurniawan", "B7789HLW", "Honda  
Revo Fit"),  
("DR4918388904", "Bagus Ade", "B7128PQR", "Honda MegaPro  
FI");
```



```

CREATE TABLE MsItem(
    ItemID CHAR(12) PRIMARY KEY NOT NULL,
    ItemName VARCHAR(255) NOT NULL,
    ItemPrice INT(100) NOT NULL
);

INSERT INTO MsItem VALUES
    ("IE94567218300", "Trio Hotdakk Mozzarella", 67000),
    ("IE84167924386", "Pure Cocoa", 26000),
    ("IE4044153406", "Baby Crab Ukuran XL", 30000),
    ("IE9041559670", "Milk Brown Sugar", 18500),
    ("IE7866181968", "Coffee Milk Brown Sugar", 20500),
    ("IE7986136188", "Martabak Upside Down Ovomaltine Keju",
145000),
    ("IE5824821466", "Martabak Tipker Coklat Kacang", 45000),
    ("IE9846391717", "Martabak Tipker Keju", 45000),
    ("IE1189800805", "Brownie Bites (Large)", 40000),
    ("IE8401724890", "Cold Fudgy Brownies 20x20", 160000);

CREATE TABLE TransactionHeader(
    OrderID CHAR(12) PRIMARY KEY NOT NULL,
    CustomerID CHAR(12) NOT NULL,
    StoreID CHAR(12) NOT NULL,
    DriverID CHAR(12) NOT NULL,
    Distance FLOAT(10) NOT NULL,
    DeliveryFee INT(10) NOT NULL,
    PaymentType VARCHAR(50) NOT NULL,
    DelivereAt time NOT NULL,
    ReceiveAt time NOT NULL,
    OrderDate date NOT NULL,

    FOREIGN KEY (CustomerID) REFERENCES
MsCustomer(CustomerID),
    FOREIGN KEY (StoreID) REFERENCES MsStore(StoreID),
    FOREIGN KEY (DriverID) REFERENCES MsDriver(DriverID)
);

INSERT INTO TransactionHeader VALUES
    ("F-1942805643", "CU6182349271", "ST8945173962",
"DR1234854718", 3.4, 13000, "Gopay", "17:43", "18:06",
"2022-12-08"),

```

```

        ("F-3697529370", "CU8200800827", "ST2241853257",
"DR4918388904", 1.9, 10000, "Gopay", "09:00", "09:40",
"2022-12-08"),
        ("F-6475451635", "CU6018155509", "ST2241853257",
"DR6841368364", 1.3, 10000, "Tunai", "10:12", "11:15",
"2022-12-08"),
        ("F-2536533461", "CU5839418311", "ST2241853257",
"DR9005452865", 1.6, 10000, "Gopay Coins", "13:15", "13:40",
"2022-12-08"),
        ("F-7093055252", "CU3872338300", "ST4241021513",
"DR7024063722", 2.3, 1500, "Gopay", "09:00", "09:34",
"2022-12-08"),
        ("F-6655195396", "CU6951981065", "ST1197623553",
"DR7006973912", 3.6, 13000, "Gopay", "10:34", "11:15",
"2022-12-08"),
        ("F-5443526038", "CU9720788482", "ST6370240598",
"DR4116616002", 4.5, 15000, "Gopay", "09:00", "09:45",
"2022-12-08"),
        ("F-1064343062", "CU6226426715", "ST6370240598",
"DR6056883237", 1.5, 10000, "Gopay", "19:00", "19:55",
"2022-12-08"),
        ("F-1612031013", "CU2529474198", "ST4241021513",
"DR5497501521", 0.8, 10000, "Gopay", "20:00", "20:25",
"2022-12-08"),
        ("F-9401563756", "CU5361522657", "ST6848506081",
"DR3109658692", 0.3, 10000, "Gopay", "11:00", "11:30",
"2022-12-08");

```

```

CREATE TABLE TransactionDetail(
    OrderID CHAR(12) NOT NULL,
    ItemID CHAR(12) NOT NULL,
    Quantity INT(10) NOT NULL,

    PRIMARY KEY (OrderID, ItemID),
    FOREIGN KEY (OrderID) REFERENCES
TransactionHeader(OrderID),
    FOREIGN KEY (ItemID) REFERENCES MsItem (itemID)
);

```

```

INSERT INTO TransactionDetail VALUES
    ("F-1942805643", "IE94567218300", 1),
    ("F-1942805643", "IE84167924386", 1),
    ("F-6475451635", "IE1189800805", 2),
    ("F-6475451635", "IE8401724890", 1),

```

```

        ("F-2536533461", "IE1189800805", 1),
        ("F-7093055252", "IE7866181968", 3),
        ("F-7093055252", "IE9041559670", 2),
        ("F-1064343062", "IE7986136188", 1),
        ("F-1064343062", "IE9846391717", 1),
        ("F-1612031013", "IE7866181968", 1);

-- tampilkan struk dari database

SELECT TH.OrderID, TH.CustomerID, TH.StoreID, TH.DriverID,
TH.Distance, TH.DeliveryFee, TH.PaymentType, TH.DelivereAt,
TH.ReceiveAt, TH.OrderDate, Item.ItemName,
Item.ItemPrice,TD.Quantity, Store.StoreName,
Store.StoreAddress, Store.ServiceAndOtherFee,
Customer.CustomerName, Customer.CustomerAddress,
Driver.DriverName, Driver.DriverLicensePlate,
Driver.DriverVehicleType, District.DistrictName AS `Store
District`, Districts.DistrictName AS `Customer District`,
provincestore.ProvinceName AS `Province Store`,
provincecustomer.ProvinceName AS `Province Customer`,
urbanstore.UrbanVillageName AS `Urban Village Store`,
urbancustomer.UrbanVillageName AS `Urban Village Customer`,
codestore.PostalCodeNumber AS `Postal Code Store`,
codecustomer.PostalCodeNumber AS `Postal Code Customer`,
subdistrictstore.SubDistrictName AS `Sub District Store`,
subdistrictcustomer.SubDistrictName AS `Sub District Cusomer`

FROM
transactionheader AS TH
JOIN
transactiondetail AS TD
ON
TH.OrderID = TD.OrderID
JOIN
msitem AS Item
ON
TD.ItemID = Item.ItemID
JOIN
msstore AS Store
ON
TH.StoreID = Store.StoreID
JOIN
mscustomer AS Customer
ON

```

```

TH.CustomerID = Customer.CustomerID
JOIN
msdriver AS Driver
ON
TH.DriverID = Driver.DriverID

-- District Store and District Customer
JOIN
msdistrict AS District
ON
Store.StoreDistrictID = District.DistrictID
JOIN
msdistrict AS Districts
ON
Districts.DistrictID = Customer.CustomerDistrictID

-- Province Store and Customer
JOIN
msprovince AS provincestore
ON
Store.StoreProvinceID = provincestore.ProvinceID
JOIN
msprovince AS provincecustomer
ON
provincecustomer.ProvinceID = Customer.CustomerProvinceID

-- Urban Village Store and Customer
JOIN
msurbanvillage AS urbanstore
ON
Store.StoreUrbanVillageID = urbanstore.UrbanVillageID
JOIN
msurbanvillage AS urbancustomer
ON
urbancustomer.UrbanVillageID = Customer.CustomerUrbanVillageID

-- Postal Code Store and Customer
JOIN
mspostalcode AS codestore
ON
Store.StorePostalCodeID = codestore.PostalCodeID
JOIN
mspostalcode AS codecustomer
ON

```

```
codecustomer.PostalCodeID = Customer.CustomerPostalCodeID
```

```
-- Sub District Store and Customer
```

```
JOIN
```

```
mssubdistrict AS subdistrictstore
```

```
ON
```

```
Store.StoreSubDistrictID= subdistrictstore.SubDistrictID
```

```
JOIN
```

```
mssubdistrict AS subdistrictcustomer
```

```
ON
```

```
subdistrictcustomer.SubDistrictID =
```

```
Customer.CustomerSubDistrictID
```

## Output

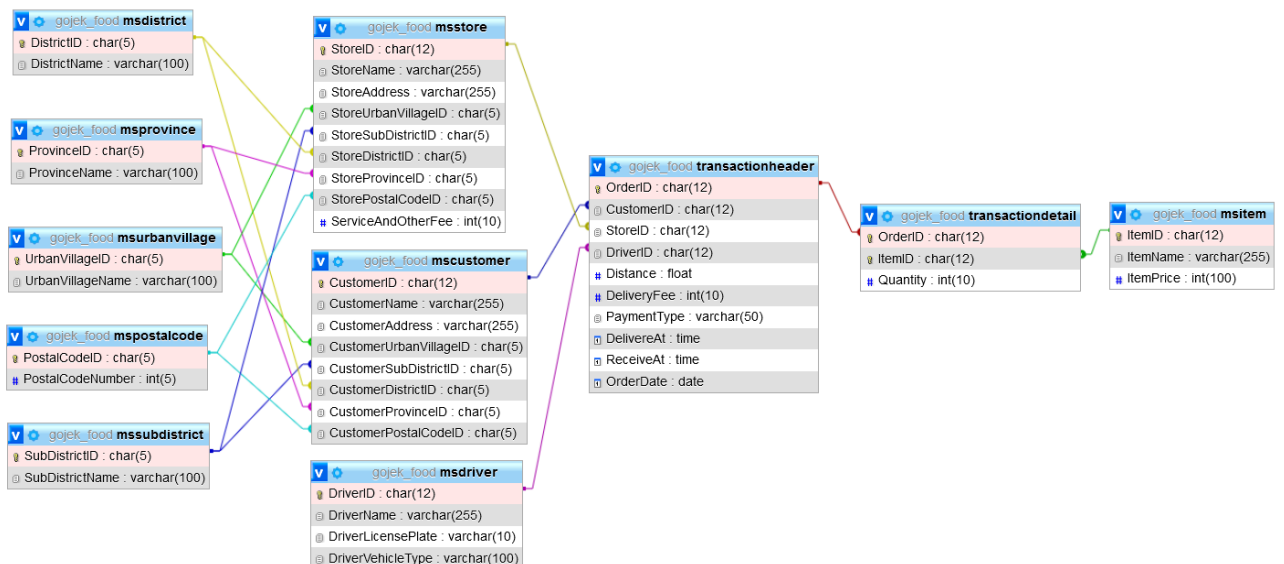
OrderID	CustomerID	StoreID	DriverID	Distance	DeliveryFee	PaymentType	DelivereAt	ReceiveAt	OrderDate	ItemName
F-1064343062	CU6226426715	ST6370240598	DR6056883237	1.5	10000	Gopay	19:00:00	19:55:00	2022-12-08	Martabak Upside Down Ovomaltine Keju
F-1064343062	CU6226426715	ST6370240598	DR6056883237	1.5	10000	Gopay	19:00:00	19:55:00	2022-12-08	Martabak Tipker Keju
F-1612031013	CU2529474198	ST4241021513	DR5497501521	0.8	10000	Gopay	20:00:00	20:25:00	2022-12-08	Coffee Milk Brown Sugar
F-1942805643	CU6182349271	ST8945173962	DR1234854718	3.4	13000	Gopay	17:43:00	18:06:00	2022-12-08	Pure Cocoa
F-1942805643	CU6182349271	ST8945173962	DR1234854718	3.4	13000	Gopay	17:43:00	18:06:00	2022-12-08	Trio Hotdakk Mozarella

ItemName	ItemPrice	Quantity	StoreName	StoreAddress	ServiceAndOtherFee	CustomerName	CustomerAddress	DriverName	DriverLicensePlate		
Martabak Upside Down Ovomaltine Keju	145000	1	Martabak Pecenongan 78	Jalan Entong Gendut Nomor 94	12000	I Ketut	Jl Melati Kav. 97	Putri Rahayu	B2208MLO		
Martabak Tipker Keju	45000	1	Martabak Pecenongan 78	Jalan Entong Gendut Nomor 94	12000	I Ketut	Jl Melati Kav. 97	Putri Rahayu	B2208MLO		
Coffee Milk Brown Sugar	20500	1	Jakarta Cheese Factory	Komplek Podomoro City, Jl. Letjen S. Parman Kavlin...	15000	Nurul Putri	Jl Tulip Blok B No. 1	Yunita Rina	B6488AAV		
Pure Cocoa	26000	1	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jalan Pamulang Permai 1	9000	Fepri Putra	Jl Mawar 3 Blok C3 No. 1	Agam Maulana	B6394WUF		
Trio Hotdakk Mozarella	67000	1	Chatime X Cupbop, LP Pamulang	Komplek Pamulang Permai, Jalan Pamulang Permai 1	9000	Fepri Putra	Jl Mawar 3 Blok C3 No. 1	Agam Maulana	B6394WUF		
DriverLicensePlate	DriverVehicleType	Store District	Customer District	Province Store	Province Customer	Urban Village Store	Urban Village Customer	Postal Code Store	Postal Code Customer	Sub District Store	Sub District Customer
B2208MLO	Honda Beat	Jakarta Timur	Karawang	Daerah Istimewa Yogyakarta	Jawa Barat	Cipete Selatan	Periuk Jaya	15415	46257	Palmerah	Palmerah
B2208MLO	Honda Beat	Jakarta Timur	Karawang	Daerah Istimewa Yogyakarta	Jawa Barat	Cipete Selatan	Periuk Jaya	15415	46257	Palmerah	Palmerah
B6488AAV	Honda Beat	Jakarta Barat	Jakarta Barat	DKI Jakarta	Jawa Barat	Silipi	Duren Sawit	15418	15415	Palmerah	Palmerah
B6394WUF	Honda Beat	Tangerang Selatan	Tangerang Selatan	Banten	Banten	Pamulang Barat	Kedaung	15417	15415	Pamulang	Pamulang
B6394WUF	Honda Beat	Tangerang Selatan	Tangerang Selatan	Banten	Banten	Pamulang Barat	Kedaung	15417	15415	Pamulang	Pamulang

OrderID	CustomerID	StoreID	DriverID	Distance	DeliveryFee	PaymentType	DelivereAt	ReceiveAt	OrderDate	ItemName
F-1942805643	CU6182349271	ST8945173962	DR1234854718	3.4	13000	Gopay	17:43:00	18:06:00	2022-12-08	Trio Hotdakk Mozarella
F-2536533461	CU5839418311	ST2241853257	DR9005452865	1.6	10000	Gopay Coins	13:15:00	13:40:00	2022-12-08	Brownie Bites (Large)
F-6475451635	CU6018155509	ST2241853257	DR6841368364	1.3	10000	Tunai	10:12:00	11:15:00	2022-12-08	Brownie Bites (Large)
F-6475451635	CU6018155509	ST2241853257	DR6841368364	1.3	10000	Tunai	10:12:00	11:15:00	2022-12-08	Cold Fudgy Brownies 20x20
F-7093055252	CU3872338300	ST4241021513	DR7024063722	2.3	1500	Gopay	09:00:00	09:34:00	2022-12-08	Coffee Milk Brown Sugar
F-7093055252	CU3872338300	ST4241021513	DR7024063722	2.3	1500	Gopay	09:00:00	09:34:00	2022-12-08	Milk Brown Sugar
ItemName	ItemPrice	Quantity	StoreName	StoreAddress	ServiceAndOtherFee	CustomerName	CustomerAddress	DriverName	DriverLicensePlate	
Trio Hotdakk Mozarella	67000	1	Cupbop, LP Pamulang	Permai, Jalan Pamulang Permai 1	9000	Fepri Putra	No. 1	Agam Maulana	B6394WUF	
Brownie Bites (Large)	40000	1	Cold Brownies Caja	Komplek Pamulang Permai, Jalan Pamulang Permai 1	15000	Puspita Ayu	Jl Mawar 2 Blok A4 No.5	Puji Devi	B1608WAE	
Brownie Bites (Large)	40000	2	Cold Brownies Caja	Komplek Podomoro City, Jl. Letjen S. Parman Kavlin...	15000	Budi Adi	Jl Melati 1 Blok A No.10	Arief Kurniawan	B7789HLW	
Cold Fudgy Brownies 20x20	160000	1	Cold Brownies Caja	Komplek Podomoro City, Jl. Letjen S. Parman Kavlin...	15000	Budi Adi	Jl Melati 1 Blok A No.10	Arief Kurniawan	B7789HLW	
Coffee Milk Brown Sugar	20500	3	Jakarta Cheese Factory	Komplek Podomoro City, Jl. Letjen S. Parman Kavlin...	15000	Novita Dyah	Jl Mawar 3 Blok C3 No. 6	Citra Munaroh	B6086BQN	
Milk Brown Sugar	18500	2	Jakarta Cheese Factory	Komplek Podomoro City, Jl. Letjen S. Parman Kavlin...	15000	Novita Dyah	Jl Mawar 3 Blok C3 No. 6	Citra Munaroh	B6086BQN	

DriverLicensePlate	DriverVehicleType	Store District	Customer District	Province Store	Province Customer	Urban Village Store	Urban Village Customer	Postal Code Store	Postal Code Customer	Sub District Store	Sub District Customer
B6394WUF	Honda Beat	Tangerang Selatan	Tangerang Selatan	Banten	Banten	Pamulang Barat	Kedaung	15417	15415	Pamulang	Pamulang
B1608WAE	Honda Beat	Jakarta Barat	Tangerang Selatan	DKI Jakarta	Banten	Slipi	Kedaung	15418	15415	Palmerah	Pamulang
B7789HLW	Honda Revo Fit	Jakarta Barat	Karawang	DKI Jakarta	Jawa Barat	Slipi	Periuk Jaya	15418	46257	Palmerah	Palmerah
B7789HLW	Honda Revo Fit	Jakarta Barat	Karawang	DKI Jakarta	Jawa Barat	Slipi	Periuk Jaya	15418	46257	Palmerah	Palmerah
B6086BQN	Honda Revo Fit	Jakarta Barat	Tangerang Selatan	DKI Jakarta	Banten	Slipi	Kedaung	15418	15415	Palmerah	Pamulang
B6086BQN	Honda Revo Fit	Jakarta Barat	Tangerang Selatan	DKI Jakarta	Banten	Slipi	Kedaung	15418	15415	Palmerah	Pamulang

## Struktur





## **BAB V**

### **KESIMPULAN**

Database adalah kumpulan data yang terorganisir dan saling berhubungan. Database memiliki fungsi penyimpanan, penghapusan, pengarsipan, pengelompokan, serta pencegahan duplikasi data. Database yang baik sangat dibutuhkan agar dapat menangani pemrosesan dengan cepat, tepat, dan efisien. Dan sebaliknya, database yang buruk akan mempersulit pengguna dalam menganalisis dan mengekstrak informasi penting dari data.

Penulis percaya bahwa grafik ERD yang telah dibuat adalah cara yang paling efisien untuk menangani database transaksi pemesanan makanan/minuman melalui aplikasi online. Tanpa adanya database transaksi yang baik, pengguna mungkin akan kesulitan untuk memantau informasi seperti jenis barang yang dibeli oleh konsumen, tanggal pembelian, informasi konsumen, jumlah barang yang dibeli, dan sebagainya. ERD yang baik sangatlah penting sebagai dasar untuk merancang sistem database yang baik pula.


Dari proyek ini kami belajar tentang bagaimana membuat model database yang baik serta bagaimana cara menormalisasi data jika disajikan struk belanja. Langkah pertama ialah memahami kepentingan dan tujuan pembuatan database, kemudian mencatat atribut apa saja yang dibutuhkan, selanjutnya melakukan normalisasi data dari model UNF hingga model BCNF jika memungkinkan. Langkah terakhir ialah membuat ERD model agar sistem database mudah dipahami dan dimengerti. Model yang kami buat merupakan salah satu alternatif yang dapat dipakai untuk membuat database transaksi minuman/makanan via aplikasi online. Model database yang kami buat memungkinkan untuk memiliki kemiripan dengan struk belanja online e-commerce lainnya.

## **DAFTAR PUSTAKA**

1. Ramez Elmasri , Shamkant Navathe (2015). Fundamentals of Database Systems 7th Edition. Part 6 Database Design Theory and Normalization, Chapter 14. Basics of Functional Dependencies and Normalization for Relational Databases, Page 489 - 521.
2. Thomas Connolly, Carolyn Begg (2015). Database Systems Sixth Edition: New York: Peason. Part 3 Database Analysis and Design, Chapter 14. Normalization, Page 451-493

## LAMPIRAN

### 1. Data struk:

 Thursday, 8 December 2022  
Order ID: F-1942805643

Hi Fepri Putra,

**Thanks for ordering GoFood**

**Total paid** **Rp115.000**

**Order details**

1 Trio Hotdakk Mozarella	@Rp67.000	<b>Rp67.000</b>
1 Pure Cocoa	@Rp26.000	<b>Rp26.000</b>
No cutlery/straws. Thanks for reducing single-use waste!		
<b>Total Price</b>		<b>Rp93.000</b>
Delivery fee		Rp13.000
Service and other fees		Rp9.000
<b>Total payment</b>		<b>Rp115.000</b>
Paid with GoPay		Rp115.000

**Delivery details**

**Agam Maulana**  
B6394WUF • Honda Beat

✓ Distance 3.4 km  
⌚ Delivery time 23 mins






📍 Delivered at 17:43 from  
**Chatime x Cupbop, LP Pamulang**  
Komplek Pamulang Permai, Jl. Pamulang Permai 1, Pamulang, Tangerang


🕒 Received at 18:06 at  
**Jl. Mawar 3 Blok C3 No.1**  
Jl. Mawar 3 Blok C3 No.1, Kedaung, Pamulang, Kota Tangerang Selatan, Banten 15415, Indonesia

[Help](#) [Report problem](#) [About GoFood](#)

The final total price is the price paid when the order is completed. It can be different from the estimated price due to stock availability or another specific reasons. The prices listed are final. Additional fees such as tips given after the order is completed are not included in this receipt.

Connect with us



Pasaraya Blok M GD B, 7th Floor, Kebayoran Baru, DKI Jakarta Indonesia 12160

### 2. PPT Presentasi:

[https://drive.google.com/drive/folders/1IVDU3QAMLcVIM0AmGC43DppVRBG\\_3gSp?usp=sharing](https://drive.google.com/drive/folders/1IVDU3QAMLcVIM0AmGC43DppVRBG_3gSp?usp=sharing)