

LAPORAN

PEMODELAN KONSEPTUAL DAN LOGICAL

Disusun untuk memenuhi salah satu tugas mata kuliah CII1J3 Pemodelan Basis Data

Oleh :

Aidil Arifin Nizar - 1301204180

Hidayat Taufiqur R.A. - 1301204300

Muhammad Rafli Susanto - 1301204052



PROGRAM STUDI INFORMATIKA

FAKULTAS INFORMATIKA

2020

PENDAHULUAN

Penulisan laporan ini bertujuan untuk menjelaskan hasil tugas kelompok Pemodelan Basis Data. Kelompok kami memilih studi kasus yang berjudul E-shop yang aturan bisnisnya sudah ditentukan.

Secara garis besar, E-Shop merupakan sebuah aplikasi marketplace yang menghubungkan seller dan buyer untuk bertransaksi jual beli barang secara online. Marketplace tersebut memungkinkan seorang user yang diidentifikasi berdasarkan alamat email dan nomor handphone menjadi seller dan buyer sekaligus.

Dari aturan bisnis E-Shop, kami menerjemahkannya menjadi sebuah ER-Diagram dan kemudian kami membuat skema relasi dari ER-Diagram yang telah kami buat. Kami juga menambahkan beberapa asumsi untuk menghilangkan ambiguitas dari aturan bisnis.

Oleh karena itu, kami membuat suatu rancangan ER-Diagram beserta skema relasinya sesuai dengan aturan bisnis yang ada, dengan tujuan untuk memudahkan proses perancangan database.

ATURAN BISNIS

- *Seller dan buyer adalah user yang diidentifikasi berdasarkan alamat email, data user yang perlu disimpan selain email adalah nomor handphone. Seorang user dapat menjadi seller dan buyer sekaligus.*
- *User yang bertindak sebagai seller memiliki nama toko, dan alamat toko.*
- *User yang bertindak sebagai buyer memiliki property nama buyer dan gender.*
- *Seller menjual barang yang diidentifikasi berdasarkan id, nama barang, deskripsi, dan harga. Tidak lupa dicatat jumlah stok yang tersedia. Product yang sama dari seller berbeda, dicatat pada record terpisah.*
- *Transaksi dapat memiliki status: belum dibayar - sudah dibayar - dikemas oleh seller - dikirim oleh seller - selesai*
- *Untuk setiap transaksi, dicatat id transaksi, alamat pengiriman, total nilai, dan detail barang yang dibeli (beserta kuantitas masing-masing varian). Total nilai merupakan penjumlahan nilai barang yang dibeli.*
- *Ongkos kirim diberikan gratis oleh E-Shop, jadi tidak perlu dihitung dan namun tetap perlu dicatat (biaya, nomor resi).*
- *Untuk setiap transaksi, buyer melakukan transfer sebagai metode pembayaran. E-Shop menyediakan beberapa rekening pada bank berbeda, sehingga buyer dapat memilih transfer melalui bank tertentu. Data pembayaran mencatat jumlah yang dibayar, nama bank tujuan, rekening tujuan, serta bank dan rekening pengirim.*
- *Untuk setiap transaksi yang sudah dibayar oleh buyer, E-Shop akan mengirimkan data order ke seller. Barang kemudian dikemas dan dikirim oleh seller dengan menginputkan resi pengiriman.*
- *Transaksi dinyatakan selesai jika barang sudah diterima buyer.*

Asumsi:

- *Seller tidak harus menjual barang (minimal menjual 0 barang)*

PEMODELAN KONSEPTUAL

Entitas Dan Atribut

Definisikan dan jelaskan entitas apa saja yang dibentuk berdasarkan aturan bisnis yang ditetapkan. Untuk setiap entitas, tuliskan atribut yang digunakan beserta tipe atribut, deskripsi, dan contohnya. Daftar entitas diisikan pada Tabel 1, sedangkan daftar atribut untuk setiap entitas diisikan pada Tabel 2 (tambahkan tabel sesuai dengan jumlah entitas yang didefinisikan).

Tabel 1 Daftar Entitas

No.	Nama Entitas	Tipe Entitas	Deskripsi
1	USER	Entitas Kuat	USER adalah pengguna yang bisa menjadi SELLER dan BUYER sekaligus. USER memiliki composite identifier yaitu komponennya alamat email dan nomor handphone.
2	SELLER	Entitas Kuat	SELLER menjual barang yang diidentifikasi berdasarkan id, nama barang, deskripsi, dan harga. Relasi antara entitas SELLER dan BARANG mengakibatkan terbentuknya sebuah deskriptif atribut yang merecord produk yang sama dari SELLER yang berbeda
3	BUYER	Entitas Kuat	BUYER memiliki atribut nama buyer dan gender. BUYER melakukan transaksi dan akan menerima barang.
4	BARANG	Entitas Kuat	BARANG dijual oleh SELLER dan dikirimkan ke BUYER setelah BUYER melakukan pembayaran. BUYER diidentifikasi berdasarkan BarangID.
5	DATA PEMBAYARAN	Entitas Kuat	DATA PEMBAYARAN suatu entitas kuat yang memiliki atribut komposit dengan komponen harga barang dan

			ongkos kirim. DATA PEMBAYARAN juga memiliki atribut lain yaitu Nama Bank Tujuan, Rekening Tujuan, dan Bank dan Rekening Pengirim.
6	TRANSAKSI	Entitas Asosiatif	TRANSAKSI adalah entitas asosiatif yang terbentuk dari relasi many-to-many dari entitas BARANG, BUYER, dan DATA PEMBAYARAN. TRANSAKSI memiliki atribut komposit dengan komponen nama jalan, kota, provinsi dan kode pos. TRANSAKSI juga memiliki atribut total nilai, detail barang dan atribut status transaksi.

Nama Entitas : Customer (*contoh*)

Tabel 2 Daftar Atribut Tabel Customer (contoh)

No.	Nama Atribut	Tipe Atribut	Deskripsi	Contoh
<i>contoh</i>	<i>ID Pelanggan</i>	<i>Simple Attribute</i>	<i>ID Pelanggan adalah nomor identitas unik yang diberikan perusahaan kepada setiap customer.</i>	<i>1234567890</i>
1	UserID	Composite Identifier	USERID berisi alamat email dan nomor handphone user	lusiusCS
2	Nama Toko	Simple Attribute	Nama Toko berisi nama toko dari user SELLER	Markop Shop
3	Alamat Toko	Composite Attribute	Alamat Toko berisi alamat lengkap toko fisik.	Jl. Pintu Surga Kondangsari, Kec. Beber, Cirebon, Jawa Barat 45172
4	Nama	Composite Attribute	Nama berisi nama dari user BUYER	Markop
5	Gender	Simple Attribute	Gender berisi jenis kelamin dari user BUYER	Laki-laki
6	BarangID	Identifier	BarangID adalah nomor identitas unik yang mengidentifikasi barang yang dijual SELLER	987654321
7	Nama Barang	Simple Attribute	Nama Barang adalah Nama barang yang akan dijual oleh user Seller	Laptop
8	Deskripsi	Simple Attribute	Deskripsi berisi penjelasan singkat dari barang yang dijual SELLER	Kami menjual laptop merk A*ple dengan spesifikasi ... dengan harga super miring!!!

9	Harga	Simple Attribute	Harga adalah suatu digit (dalam rupiah) yang merepresentasikan nilai dari suatu barang yang dijual oleh seller.	10000
10	Stok	Simple Attribute	Stok adalah suatu digit yang merepresentasikan jumlah sisa barang yang dimiliki oleh SELLER.	5
11	Record Barang Yang Sama	Descriptive Attribute	Atribut ini terbuat dari relasi SELLER dengan BARANG yang bertujuan untuk merecord produk yang sama dari SELLER yang berbeda.	
12	OrderID	Identifier	OrderID adalah nomor identifikasi unik yang bertujuan untuk mengidentifikasi pesanan yang diproses oleh SELLER dan yang dilakukan oleh BUYER	1243218374
13	Jumlah Pembayaran	Composite Attribute	Jumlah Pembayaran adalah harga total yang harus ditransaksikan oleh BUYER	420690
14	Nama Bank Tujuan	Simple Attribute	Nama Bank Tujuan adalah nama bank yang akan dituju oleh BUYER saat melakukan pembayaran	BJB
15	Rekening tujuan	Simple Attribute	Rekening tujuan adalah rekening bank tempat BUYER mentransfer dana	0990118118
16	Bank dan Rekening Pengirim	Simple Attribute	Bank dan Rekening Pengirim berisi Nama Bank dan Rekening Pengirim yang sudah melakukan transaksi ke rekening tujuan	BNI, 1670000677335
17	Alamat Pengiriman	Composite Attribute	Alamat Pengiriman berisi alamat tujuan pengiriman barang yang ditentukan oleh BUYER	Jl. Shirothol Mustaqim RT9, RW10, Utan Kayu Utara, Kec. Matraman, Jakarta Timur, Jakarta.
18	Total Nilai	Simple Attribute	Total Nilai adalah Jumlah pembayaran yang harus dibayarkan oleh BUYER dan terlampir pada sebuah transaksi pembayaran.	690690
19	Detail Barang	Composite Attribute	Detail Barang adalah ringkasan dari detail	Tipe: Laptop Berat: 2KG

			barang yang dipesan oleh BUYER	Kondisi: Baru
20	TransaksiID	Identifier	TransaksiID adalah nomor identifikasi unik yang bertujuan untuk mengidentifikasi transaksi yang dilakukan oleh BUYER.	AB12345678665
21	Status Transaksi	Simple Attribute	Status Transaks berisii keterangan transaksi berupa seperti: belum dibayar - sudah dibayar - dikemas oleh seller - dikirim oleh seller - selesai	selesai

Relasi

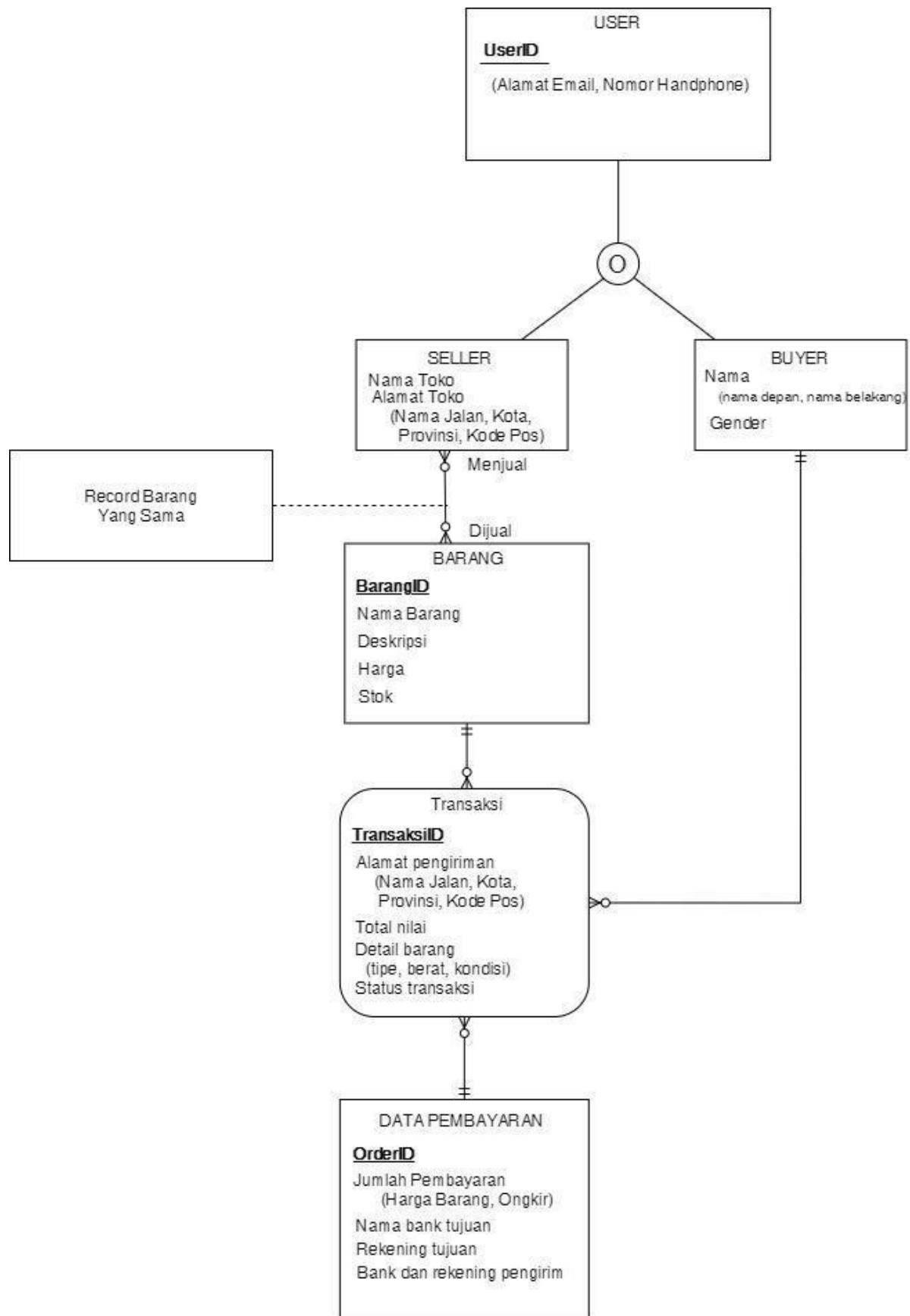
Tuliskan nama relasi dan kardinalitas yang terbentuk antar entitasnya. Untuk relasi binary, silakan isikan pada tabel 3. Silakan sesuaikan tabelnya jika diperlukan pendefinisian relasi unary atau ternary.

Tabel 3 Daftar Relasi

No.	Nama Entitas #1	Nama Entitas #2	Nama Relasi	Kardinalitas	Constraint
1	USER	SELLER	Supertype/Subtype		Overlap Rule
2	USER	BUYER	Supertype/Subtype		Overlap Rule
3	SELLER	BARANG	Menjual	Many-to-many	Optional
4	BARANG	SELLER	Dijual	Many-to-many	Optional
5	BUYER	TRANSAKSI	Asosiatif	One-to-many	Mandatory
6	TRANSAKSI	BUYER	Asosiatif	Many-to-one	Optional
7	BARANG	TRANSAKSI	Asosiatif	One-to-many	Mandatory
8	TRANSAKSI	BARANG	Asosiatif	Many-to-one	Optional
9	TRANSAKSI	DATA PEMBAYARAN	Asosiatif	Many-to-one	Optional
10	DATA PEMBAYARAN	TRANSAKSI	Asosiatif	One-to-many	Mandatory

E-R Diagram

Gambarkan E-R diagram sesuai dengan pendefinisian entitas, atribut, dan relasi pada Gambar 1.

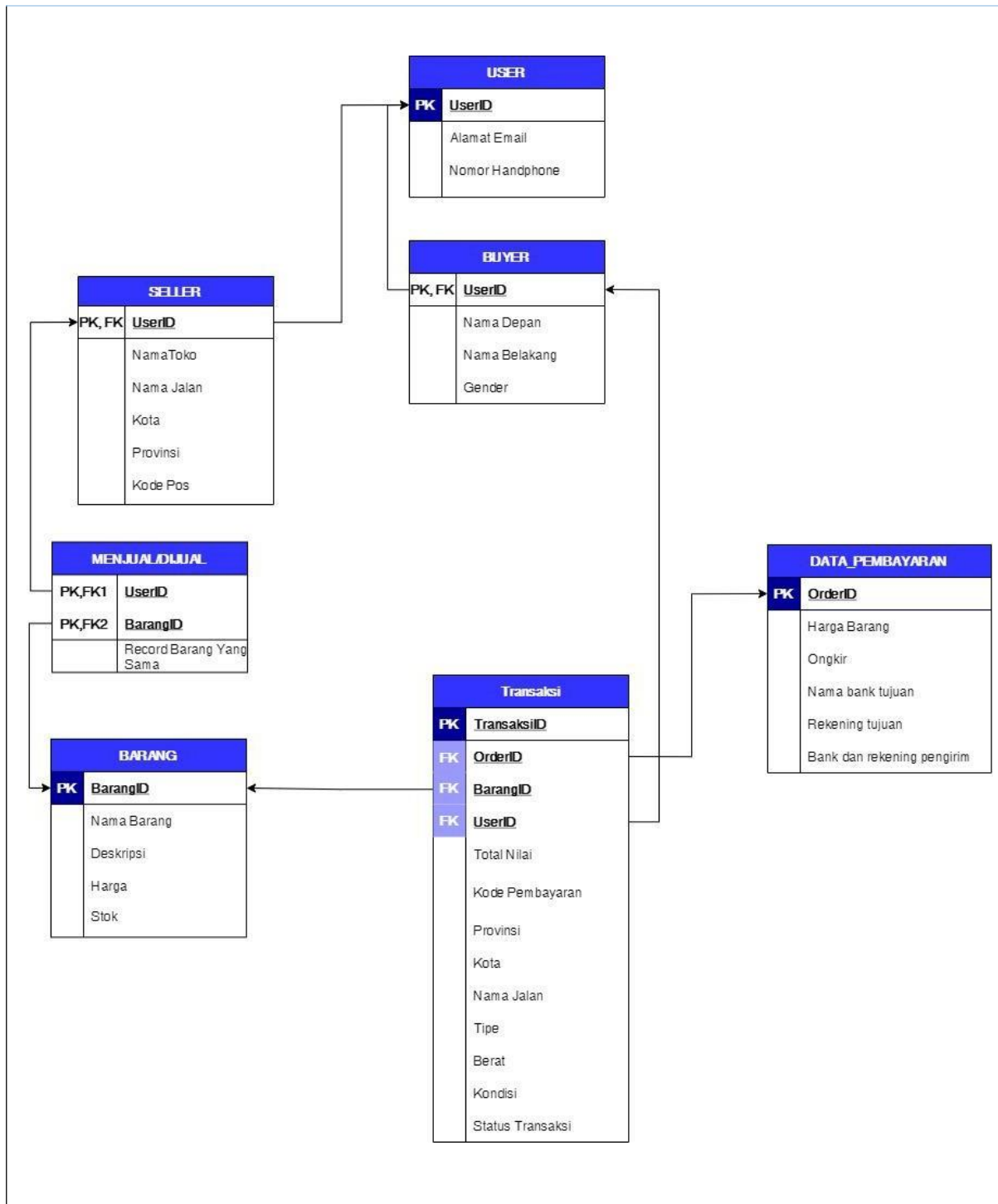


Gambar 1 E-R Diagram

PEMODELAN LOGICAL

Skema Relasi

Petakan E-R Diagram yang sudah digambarkan pada bagian pemodelan konseptual ke dalam skema relasi (Gambar 2). Perhatikan setiap tipe entitas, tipe atribut, dan kardinalitas dalam melakukan pemetaan. Sertakan Primary Key dan Foreign Key (jika ada) untuk setiap relasi yang dihasilkan.



Gambar 2 Skema Relasi

Physical Database Design

Tabel User

Column_name	Data type	Length/size	Constraint	Reference
UserID	varchar	25	Primary Key	-
AlamatEmail	varchar	99	Not Null	-
NomorHP	varchar	13	Not Null	-

Tabel Buyer

Column_name	Data type	Length/size	Constraint	Reference
UserID	varchar	25	Primary Key Foreign Key	User(UserID)
NamaDepan	varchar	25	Not Null	-
NamaBelakang	varchar	25	Not Null	-
Gender	char	1	Not Null	-

Tabel Seller

Column_name	Data type	Length/size	Constraint	Reference
UserID	varchar	25	Primary Key Foreign Key	User(UserID)
NamaToko	varchar	25	Not Null	-
NamaJalan	varchar	99	Not Null	-
Kota	varchar	25	Not Null	-
Provinsi	varchar	25	Not Null	-
KodePos	char	5	Not Null	-

Tabel Barang

Column_name	Data type	Length/size	Constraint	Reference
BarangID	char	10	Primary Key	-
NamaBarang	varchar	25	Not null	-
Deskripsi	varchar	1000	Not null	-
Harga	Bigint unsigned	25	Not null	-
Stok	Int unsigned	0-4294967295	Not null	-

Tabel Menjual/Dijual

Column_name	Data type	Length/size	Constraint	Reference
UserID	varchar	25	Foreign Key	User(UserID)
BarangID	char	10	Foreign Key	Barang(BarangID)
RecordBarangYgS ama	varchar	1000	Not null	-

Tabel Data_Pembayaran

Column_name	Data type	Length/size	Constraint	Reference
OrderID	char	10	Primary Key	-
HargaBarang	Bigint unsigned	25	Not Null	-
Ongkir	Int unsigned	25	Not Null	-
NamaBankTujuan	varchar	25	Not Null	-
RekeningTujuan	varchar	25	Not Null	-
BankRekeningPen girim	varchar	25	Not Null	-

Tabel Transaksi

Column_name	Data type	Length/size	Constraint	Reference
TransaksiID	char	10	Primary Key	-
OrderID	char	10	Foreign Key	Data_Pembayaran(OrderID)
BarangID	char	10	Foreign Key	Barang(BarangID)
UserID	varchar	25	Foreign Key	User(UserID)
TotalNilai	Bigint unsigned	25	Not Null	-
KodePembayaran	char	10	Not Null	-
Provinsi	varchar	100	Not Null	-
Kota	varchar	100	Not Null	-
NamaJalan	varchar	100	Not Null	-
Tipe	varchar	100	Not Null	-
Berat	Int unsigned	10	Not Null	-
Kondisi	varchar	5	Not Null	-
StatusTransaksi	varchar	30	Not Null	-

IMPLEMENTASI SQL

DDL

Masukkan script DDL yang telah dibuat beserta print screen hasil eksekusinya.

Pastikan DDL yang dieksekusi sesuai dengan skema relasi.

```
CREATE DATABASE E_shop;
```

```
Use E_shop;
```

```
CREATE TABLE User
```

```
(UserID varchar(25) PRIMARY KEY,
```

```
AlamatEmail varchar(99) not null,
```

```
NomorHP varchar(13) not null);
```

```
CREATE TABLE Buyer
```

```
(UserID varchar(25) PRIMARY KEY,
```

```
NamaDepan varchar(25) not null,
```

```
NamaBelakang varchar(25) not null,
```

```
Gender char(1) not null,
```

```
CONSTRAINT FK_UserID FOREIGN KEY (UserID)
```

```
REFERENCES User(UserID));
```

```
CREATE TABLE Seller
```

```
(UserID varchar(25) PRIMARY KEY,
```

```
NamaToko varchar(25) not null,
```

```
NamaJalan varchar(99) not null,
```

```
Kota varchar(25) not null,
```

```
Provinsi varchar(25) not null,
```

```
KodePos char(5) not null,
```

```
CONSTRAINT FK_UserID_1 FOREIGN KEY (UserID)
```

```
REFERENCES User(UserID));
```

```
CREATE TABLE Barang
```

```
(BarangID char(10) PRIMARY KEY,
```

```
NamaBarang varchar(25) not null,
```

```
Deskripsi varchar(1000) not null,
```

```
Harga bigint(25) unsigned not null,
```

```
Stok int unsigned not null);
```

```
CREATE TABLE Menjual_Dijual
(UserID varchar(25),
BarangID char(10),
RecordBarangYgSama varchar(1000) not null,
CONSTRAINT FK_UserID_2 FOREIGN KEY (UserID)
REFERENCES User(UserID),
CONSTRAINT FK_BarangID FOREIGN KEY (BarangID)
REFERENCES Barang(BarangID));
```

```
CREATE TABLE Data_Pembayaran
(OrderID char(10) PRIMARY KEY,
HargaBarang bigint(25) unsigned not null,
Ongkir int(25) unsigned not null,
NamaBankTujuan varchar(25) not null,
RekeningTujuan varchar(25) not null,
BankRekeningPengirim varchar(25) not null);
```

```
CREATE TABLE Transaksi
(TransaksiID char(10) PRIMARY KEY,
OrderID char(10),
BarangID char(10),
UserID varchar(25),
TotalNilai bigint(25) unsigned not null,
KodePembayaran char(10) not null,
Provinsi varchar(100) not null,
Kota varchar(100) not null,
NamaJalan varchar(100) not null,
Tipe varchar(100) not null,
Berat int(10) unsigned not null,
Kondisi varchar(5) not null,
StatusTransaksi varchar(30) not null,
CONSTRAINT FK_OrderID FOREIGN KEY (OrderID)
REFERENCES Data_Pembayaran(OrderID),
```

CONSTRAINT FK_BarangID_1 FOREIGN KEY (BarangID)

REFERENCES Barang(BarangID),

CONSTRAINT FK_UserID_3 FOREIGN KEY (UserID)

REFERENCES User(UserID));

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'classicmodels' expanded. The main editor window shows a SQL script with the following content:

```
1 CREATE DATABASE E_shop;
2 Use E_shop;
3
4 CREATE TABLE User
5 (UserID varchar(25) PRIMARY KEY,
6 AlamatEmail varchar(99) not null,
7 NomorHP varchar(13) not null);
8
9 CREATE TABLE Buyer
10 (UserID varchar(25) PRIMARY KEY,
11 NamaDepan varchar(25) not null,
12 NamaBelakang varchar(25) not null,
13 Gender char(1) not null,
14 CONSTRAINT FK_UserID FOREIGN KEY (UserID)
15 REFERENCES User(UserID));
16
17 CREATE TABLE Seller
18 (UserID varchar(25) PRIMARY KEY,
19 NamaToko varchar(25) not null,
20 NamaTalan varchar(99) not null);
```

The right sidebar shows a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

The bottom panel shows the 'Output' tab with the following execution results:

#	Time	Action	Message	Duration / Fetch
7	02:28:50	CREATE TABLE Barang (BarangID char(10) PRIMARY KEY, NamaBarang varchar(25) no...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be remov...	0.047 sec
8	02:28:50	CREATE TABLE Menjual_Dijual (UserID varchar(25), BarangID char(10), RecordBarangY...	0 row(s) affected	0.062 sec
9	02:28:50	CREATE TABLE Data_Pembayaran (OrderID char(10) PRIMARY KEY, HargaBarang bigi...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be remov...	0.047 sec
10	02:28:50	CREATE TABLE Transaksi (TransaksiID char(10) PRIMARY KEY, OrderID char(10), Bara...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be remov...	0.094 sec

DML

Masukkan script DML yang telah dibuat untuk setiap tabel beserta print screen hasil eksekusinya.

-- USER

INSERT INTO user

(UserID, AlamatEmail, NomorHP)

VALUES

("LusiusCS", "LusiusCS@email.com", "08123456789");

INSERT INTO user

(UserID, AlamatEmail, NomorHP)

VALUES

("Abloo", "Abloo@email.com", "08696969691");

INSERT INTO user

(UserID, AlamatEmail, NomorHP)

VALUES

("CilokGoreng", "CilokGoreng@email.com", "08142042072");

INSERT INTO user

(UserID, AlamatEmail, NomorHP)

VALUES

("TahuBulat", "TahuBulat@email.com", "08157042069");

INSERT INTO user

(UserID, AlamatEmail, NomorHP)

VALUES

("CirengAyam", "CirengAyam@email.com", "08157042062");

-- BUYER

INSERT INTO buyer

(UserID, NamaDepan, NamaBelakang, Gender)

VALUES

("LusiusCS", "Lusius", "CS", "F");

INSERT INTO buyer

```
(UserID, NamaDepan, NamaBelakang, Gender)
```

```
VALUES
```

```
("Abloo", "Abloo", "Gulugulu", "M");
```

```
INSERT INTO buyer
```

```
(UserID, NamaDepan, NamaBelakang, Gender)
```

```
VALUES
```

```
("CilokGoreng", "Cilok", "Gorenk", "M");
```

```
INSERT INTO buyer
```

```
(UserID, NamaDepan, NamaBelakang, Gender)
```

```
VALUES
```

```
("TahuBulat", "Tahu", "Bulet", "F");
```

```
INSERT INTO buyer
```

```
(UserID, NamaDepan, NamaBelakang, Gender)
```

```
VALUES
```

```
("CirengAyam", "Cireng", "Ayam", "F");
```

```
-- SELLER
```

```
INSERT INTO seller
```

```
(UserID, NamaToko, NamaJalan, Kota, Provinsi, KodePos)
```

```
VALUES
```

```
("LusiusCS", "Markop Shop", "Jl. Pintu Surga Kondangsari", "Cirebon", "Jawa Barat", "45172");
```

```
INSERT INTO seller
```

```
(UserID, NamaToko, NamaJalan, Kota, Provinsi, KodePos)
```

```
VALUES
```

```
("Abloo", "Moe Shop", "Brooklyn", "Kalimalang", "Jujutsu", "69420");
```

```
INSERT INTO seller
```

```
(UserID, NamaToko, NamaJalan, Kota, Provinsi, KodePos)
```

```
VALUES
```

```
("CilokGoreng", "Polyamory", "Lebanon Street", "Niagara", "Netanyahu", "55555");
```

```
INSERT INTO seller
```

```
(UserID, NamaToko, NamaJalan, Kota, Provinsi, KodePos)
```

```
VALUES
```

```
("TahuBulat", "Slipknot", "Jl. Asia Afrika", "Bandung", "Jawa Barat", "25022");
```

```
INSERT INTO seller
```

```
(UserID, NamaToko, NamaJalan, Kota, Provinsi, KodePos)
```

```
VALUES
```

```
("CirengAyam", "Seven Heaven", "District 7. Bekasi", "Margonda", "Jawa Barat", "17701");
```

```
-- BARANG
```

```
INSERT INTO barang
```

```
(BarangID, NamaBarang, Deskripsi, Harga, Stok)
```

```
VALUES
```

```
("1234567890", "Laptop", "Laptop Asus", 12000000, 10);
```

```
INSERT INTO barang
```

```
(BarangID, NamaBarang, Deskripsi, Harga, Stok)
```

```
VALUES
```

```
("1234567891", "Mouse Rexus", "Mouse Gaming Murah Langsung Jago", 100000, 10);
```

```
INSERT INTO barang
```

```
(BarangID, NamaBarang, Deskripsi, Harga, Stok)
```

```
VALUES
```

```
("1234567892", "Steelseries QcK+", "Mousepad Gaming Murah Langsung Jago", 200000, 10);
```

```
INSERT INTO barang
```

```
(BarangID, NamaBarang, Deskripsi, Harga, Stok)
```

```
VALUES
```

```
("1234567893", "Keyboard Corsair", "Keyboard RGB Gaming Murah Langsung Jago", 2000000, 10);
```

```
INSERT INTO barang
```

```
(BarangID, NamaBarang, Deskripsi, Harga, Stok)
```

```
VALUES
```

```
("1234567894", "RTX 3090", "VGA Ghaib", 4000000, 10);
```

```
-- DATA PEMBAYARAN
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("ABCEFG0123", 12000000, 50000, "BNI", "42069721", "BCA 42069720");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("ABCEFG0124", 100000, 22000, "BRI", "42069888", "BRI 42069770");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("ABCEFG0125", 200000, 20000, "BNI", "42069212", "BTN 42069710");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("ABCEFG0126", 2000000, 20000, "BTN", "42069212", "BCA 42069730");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("ABCEFG0127", 4000000, 20000, "BNI", "21269720", "BRI 42069520");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

VALUES

("ABCEFG0128", 12000000, 50000, "BNI", "42063720", "BCA 42069220");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("ABCEFG0129", 100000, 20000, "BNI", "42069720", "BRI 42069170");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("ABCEFG0130", 200000, 20000, "BNI", "42069720", "BTN 41069710");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("ABCEFG0131", 2000000, 20000, "BNI", "42069720", "BCA 42059730");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("ABCEFG0132", 4000000, 20000, "BNI", "42069720", "BRI 42066520");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("KOKK3OK398", 12000000, 20000, "BNI", "47129814", "BCA 42069720 ");

INSERT INTO data_pembayaran

(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)

VALUES

("AODKEOA872", 100000, 40000, "BJB", "72000342", "Mandiri 19823214");

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("VADE209122", 200000, 10000, "CIMB NIAGA", "10930232", "BRI 23190892 ");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("VR04672036", 2000000, 25000, "MANDIRI", "49002846", "BJB 77343431 ");
```

```
INSERT INTO data_pembayaran
```

```
(OrderID, HargaBarang, Ongkir, NamaBankTujuan, RekeningTujuan, BankRekeningPengirim)
```

```
VALUES
```

```
("00D9EAA331", 4000000, 100000, "BCA", "65934371", "BNI 32183122 ");
```

```
-- RECORD BARANG YANG SAMA
```

```
INSERT INTO menjual_dijual
```

```
(UserID, BarangID, RecordBarangYgSama)
```

```
VALUES
```

```
("LusiusCS", "1234567890", "Laptop");
```

```
INSERT INTO menjual_dijual
```

```
(UserID, BarangID, RecordBarangYgSama)
```

```
VALUES
```

```
("CirengAyam", "1234567890", "Laptop");
```

```
INSERT INTO menjual_dijual
```

```
(UserID, BarangID, RecordBarangYgSama)
```

```
VALUES
```

```
("Abloo", "1234567890", "Laptop");
```

```
INSERT INTO menjual_dijual
```

```
(UserID, BarangID, RecordBarangYgSama)
```

VALUES

("Abluu", "1234567891", "Mouse Rexus");

INSERT INTO menjual_dijual

(UserID,BarangID,RecordBarangYgSama)

VALUES

("CirengAyam", "1234567892", "Steelseries Qck+");

INSERT INTO menjual_dijual

(UserID,BarangID,RecordBarangYgSama)

VALUES

("TahuBulat", "1234567893", "Keyboard Corsair");

INSERT INTO menjual_dijual

(UserID,BarangID,RecordBarangYgSama)

VALUES

("CilokGoreng", "1234567894", "RTX 3090");

-- TRANSAKSI

-- 1

INSERT INTO transaksi

(TransaksiID,OrderID,BarangID,UserID,TotalNilai,

KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,

Berat,Kondisi,StatusTransaksi)

VALUES

("01234ABCDE", "ABCEFG0128", "1234567890", "LusiusCS", 12020000,

"AB010", "Jawa Barat", "Cirebon", "Jl. Pintu Surga Kondangsari",

"Barang Elektronik", 3000, "Baru", "Belum Dibayar"

);

-- 2

INSERT INTO transaksi


```
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("11234ABCDE", "ABCEFG0129", "1234567891", "CirengAyam", 120000,  
"AB011", "Jawa Barat", "Margonda", "District 7. Bekasi",  
"Barang Elektronik", 500, "Baru", "Dikemas oleh Seller"  
);
```

-- 3

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("21234ABCDE", "ABCEFG0129", "1234567892", "Abluu", 220000,  
"AB012", "Jujutsu", "Kalimalang", "Brooklyn",  
"Barang Elektronik", 100, "Baru", "Selesai"  
);
```

-- 4

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("31234ABCDE", "ABCEFG0131", "1234567893", "Abluu", 2020000,  
"AB013", "Jujutsu", "Kalimalang", "Brooklyn",  
"Barang Elektronik", 1000, "Baru", "Sudah Dibayar"  
);
```

-- 5

```
INSERT INTO transaksi
```

```
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("41234ABCDE", "ABCEFG0132", "1234567894", "CilokGoreng", 4020000,  
"AB014", "Netanyahu", "Niagara", "Lebanon Street",  
"Barang Elektronik", 1500, "Baru", "Dikirim oleh Seller"  
);
```

-- 6

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("51234ABCDE", "ABCEFG0123", "1234567890", "LusiusCS", 12050000,  
"AB123", "Jl. Pintu Surga Kondangsari", "Cirebon", "Jawa Barat",  
"Barang Elektronik", 3000, "Baru", "Belum Dibayar"  
);
```

-- 7

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("DE3DDK33F2", "KOKK3OK398", "1234567890", "LusiusCS", 12020000,  
"FB555", "Jawa Barat", "Cirebon", "Jl. Pintu Surga Kondangsari",  
"Barang Elektronik", 3000, "Baru", "Selesai"  
);
```

-- 8

```
INSERT INTO transaksi
```

```
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("9SD34013UJ", "AODKEOA872", "1234567891", "Abloo", 140000,  
"CD729", "Jujutsu", "Kalimalang", "Brooklyn",  
"Barang Elektronik", 500, "Baru", "Sudah Dibayar"  
);
```

-- 9

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("E3D4MD4AD4", "VADE209122", "1234567892", "CilokGoreng", 210000,  
"980BH", "Netanyahu", "Niagara", "Lebanon Street",  
"Barang Elektronik", 100, "Baru", "Dikemas oleh Seller"  
);
```

-- 10

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("UPD2D34U8D", "VR04672036", "1234567893", "TahuBulat", 2025000,  
"DGU92", "Jawa Barat", "Bandung", "Jl. Asia Afrika",  
"Barang Elektronik", 1000, "Baru", "Belum Dibayar"  
);
```

-- 11

```
INSERT INTO transaksi
```

```
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("9876543219", "ABCEFG0123", "1234567890", "LusiusCS", 12050000,  
"AB123", "Jl. Pintu Surga Kondangsari", "Cirebon", "Jawa Barat",  
"Barang Elektronik", 3000, "Baru", "Belum Dibayar"  
);
```

-- 12

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("9876543218", "ABCEFG0124", "1234567891", "Abluu", 122000,  
"AB124", "Jujutsu", "Kalimalang", "Brooklyn",  
"Barang Elektronik", 500, "Baru", "Belum Dibayar"  
);
```

-- 13

```
INSERT INTO transaksi  
(TransaksiID,OrderID,BarangID,UserID,TotalNilai,  
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,  
Berat,Kondisi>StatusTransaksi)  
VALUES  
("9876543217", "ABCEFG0124", "1234567892", "CilokGoreng", 220000,  
"AB125", "Netanyahu", "Niagara", "Lebanon Street",  
"Barang Elektronik", 100, "Baru", "Belum Dibayar"  
);
```

-- 14

```
INSERT INTO transaksi
```

```

(TransaksiID,OrderID,BarangID,UserID,TotalNilai,
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,
Berat,Kondisi>StatusTransaksi)
VALUES
("9876543216", "ABCEFG0125", "1234567893", "TahuBulat", 2020000,
"AB126", "Jawa Barat", "Bandung", "Jl. Asia Afrika",
"Barang Elektronik", 1000, "Baru", "Belum Dibayar"
);

```

-- 15

```
INSERT INTO transaksi
```

```

(TransaksiID,OrderID,BarangID,UserID,TotalNilai,
KodePembayaran,Provinsi,Kota>NamaJalan,Tipe,
Berat,Kondisi>StatusTransaksi)
VALUES
("9876543215", "ABCEFG0126", "1234567894", "CirengAyam", 4020000,
"AB127", "Jawa Barat", "Margonda", "District 7. Bekasi",
"Barang Elektronik", 1500, "Baru", "Belum Dibayar"
);

```

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view of databases including 'sakila', 'sys', and 'world'. The main editor window shows a SQL script with the following content:

```

66 REFERENCES Data_Pembayaran(OrderID),
67 CONSTRAINT FK_BarangID_1 FOREIGN KEY (BarangID)
68 REFERENCES Barang(BarangID),
69 CONSTRAINT FK_UserID_3 FOREIGN KEY (UserID)
70 REFERENCES User(UserID));
71
72
73 -- USER
74 INSERT INTO user
75 (UserID, AlamatEmail, NomorHP)
76 VALUES
77 ("LusiusCS", "LusiusCS@email.com", "08123456789");
78
79 INSERT INTO user
80 (UserID, AlamatEmail, NomorHP)
81 VALUES
82 ("Abloo", "Abloo@email.com", "08696969691");
83
84 INSERT INTO user
85 (UserID, AlamatEmail, NomorHP)

```

The 'Output' panel at the bottom shows the execution results of the SQL script:

#	Time	Action	Message	Duration / Fetch
64	02:35:24	INSERT INTO transaksi (TransaksiID,OrderID,BarangID,UserID,TotalNilai, KodePembayar...	1 row(s) affected	0.016 sec
65	02:35:24	INSERT INTO transaksi (TransaksiID,OrderID,BarangID,UserID,TotalNilai, KodePembayar...	1 row(s) affected	0.000 sec
66	02:35:24	INSERT INTO transaksi (TransaksiID,OrderID,BarangID,UserID,TotalNilai, KodePembayar...	1 row(s) affected	0.000 sec
67	02:35:24	INSERT INTO transaksi (TransaksiID,OrderID,BarangID,UserID,TotalNilai, KodePembayar...	1 row(s) affected	0.000 sec

SELECT *

FROM barang;

The screenshot shows the MySQL Workbench interface with the 'e-shop' database selected. The 'barang' table is highlighted in the Navigator. The SQL editor contains the query `select * from barang`. The Result Grid displays the following data:

BarangID	NamaBarang	Deskripsi	Harga	Stok
1234567890	Laptop	Laptop Asus	12000000	10
1234567891	Mouse Rexus	Mouse Gaming Murah Langsung Jago	1000000	10
1234567892	Steelseries QcK+	Mousepad Gaming Murah Langsung Jago	2000000	10
1234567893	Keyboard Corsair	Keyboard RGB Gaming Murah Langsung Jago	2000000	10
1234567894	RTX 3090	VGA Ghaib	4000000	10

The Action Output pane shows the following results:

#	Time	Action	Message	Duration / Fetch
68	19:33:54	use e_shop	0 row(s) affected	0.000 sec
69	19:34:03	select * from barang LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SELECT *

FROM buyer;

The screenshot shows the MySQL Workbench interface with the 'e-shop' database selected. The 'buyer' table is highlighted in the Navigator. The SQL editor contains the query `select * from buyer`. The Result Grid displays the following data:

UserID	NamaDepan	NamaBelakang	Gender
Abloo	Abloo	Gulugulu	M
CikloGoreng	Ciklo	Gorenk	M
CirengAyam	Cireng	Ayam	F
LulusCS	Lulus	CS	F
TahuBulat	Tahu	Bulet	F

The Action Output pane shows the following results:

#	Time	Action	Message	Duration / Fetch
69	19:34:03	select * from barang LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
70	19:34:33	select * from buyer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SELECT *
FROM data_pembayaran;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the query: `select * from data_pembayaran`. The left sidebar shows the database schema with the `data_pembayaran` table selected. The bottom pane displays the query results in a grid format.

Table: data_pembayaran

Columns:

- OrderID: char(10), PK
- HargaBarang: bigint UN
- Ongkir: int UN
- NamaBankTujuan: varchar(2)
- RekeningTujuan: varchar(2)
- BankRekeningPengirim: varchar(2)

Result Grid:

OrderID	HargaBarang	Ongkir	NamaBankTujuan	RekeningTujuan	BankRekeningPengirim
00D9EAA331	4000000	100000	BCA	65934371	BNI 32183122
ABCEFG0123	12000000	50000	BNI	42069721	BCA 42069720
ABCEFG0124	100000	22000	BRI	42069888	BRI 42069770
ABCEFG0125	200000	20000	BNI	42069212	BTN 42069710
ABCEFG0126	2000000	20000	BTN	42069212	BCA 42069730
ABCEFG0127	4000000	20000	BNI	21269720	BRI 42069520
ABCEFG0128	12000000	50000	BNI	42063720	BCA 42069220
ABCEFG0129	100000	20000	BNI	42069720	BRI 42069170
ABCEFG0130	200000	20000	BNI	42069720	BTN 41069710
ABCEFG0131	2000000	20000	BNI	42069720	BCA 42069730
ABCEFG0132	4000000	20000	BNI	42069720	BRI 42066520
A00KE0A872	100000	40000	BJB	72000342	Mandiri 19823214

Output:

#	Time	Action	Message	Duration / Fetch
70	19:34:33	select * from buyer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
71	19:35:06	select * from data_pembayaran LIMIT 0, 1000	15 row(s) returned	0.015 sec / 0.000 sec

SELECT *
FROM menjual_dijual;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the query: `select * from menjual_dijual`. The left sidebar shows the database schema with the `menjual_dijual` table selected. The bottom pane displays the query results in a grid format.

Table: menjual_dijual

Columns:

- UserID: varchar(25)
- BarangID: char(10)
- RecordBarangYgSama: varchar(1000)

Result Grid:

UserID	BarangID	RecordBarangYgSama
LulusCS	1234567890	Laptop
CirengAyam	1234567890	Laptop
Abloo	1234567890	Laptop
Abloo	1234567891	Mouse Rexus
CirengAyam	1234567892	Steelseries Qck+
TahuBulat	1234567893	Keyboard Corsair
CikokGoreng	1234567894	RTX 3090

Output:

#	Time	Action	Message	Duration / Fetch
72	19:35:35	select * from menjual_dijual LIMIT 0, 1000	7 row(s) returned	0.016 sec / 0.000 sec
73	19:36:30	select * from menjual_dijual LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

SELECT *

FROM seller;

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'e-shop' database schema, with the 'seller' table selected. The 'Table: seller' information is shown below the navigator, listing columns: UserID (varchar(25) PK), NamaToko (varchar(25)), NamaJalan (varchar(99)), Kota (varchar(25)), Provinsi (varchar(25)), and KodePos (char(5)). The main query editor contains the SQL statement: `select * from seller`. The 'Result Grid' shows the execution results with columns: UserID, NamaToko, NamaJalan, Kota, Provinsi, and KodePos. The 'Output' pane at the bottom shows the execution log with two rows of results.

UserID	NamaToko	NamaJalan	Kota	Provinsi	KodePos
Abloo	Moe Shop	Brooklyn	Kalimang	Jujutsu	69420
ClokGoreng	Polyamory	Lebanon Street	Niagara	Netanyahu	55555
CirengAyam	Seven Heaven	District 7, Bekasi	Margonda	Jawa Barat	17701
LusiusCS	Markop Shop	Jl. Pintu Surga Kondangsari	Cirebon	Jawa Barat	45172
TahuBulat	Slipknot	Jl. Asia Afrika	Bandung	Jawa Barat	25022

Output:

#	Time	Action	Message	Duration / Fetch
73	19:36:30	select * from penjual_dijual LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
74	19:36:43	select * from seller LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

SELECT *

FROM transaksi;

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'e-shop' database schema, with the 'transaksi' table selected. The 'Table: transaksi' information is shown below the navigator, listing columns: TransaksiID (char(10) PK), OrderID (char(10)), BarangID (char(10)), UserID (varchar(25)), TotalNilai (bigint), KodePembayaran (char(10)), Provinsi (varchar(100)), Kota (varchar(100)), NamaJalan (varchar(100)), Tipe (int), Berat (int), Kondisi (varchar(5)), and StatusTransaksi (varchar(30)). The main query editor contains the SQL statement: `select * from transaksi`. The 'Result Grid' shows the execution results with columns: TransaksiID, OrderID, BarangID, UserID, TotalNilai, KodePembayaran, Provinsi, and Kota. The 'Output' pane at the bottom shows the execution log with two rows of results.

TransaksiID	OrderID	BarangID	UserID	TotalNilai	KodePembayaran	Provinsi	Kota
01234ABCDE	ABCEFG0128	1234567890	LusiusCS	12020000	AB010	Jawa Barat	Cirebon
11234ABCDE	ABCEFG0129	1234567891	CirengAyam	1200000	AB011	Jawa Barat	Margonda
21234ABCDE	ABCEFG0129	1234567892	Abloo	2200000	AB012	Jujutsu	Kalimang
31234ABCDE	ABCEFG0131	1234567893	Abloo	20200000	AB013	Jujutsu	Kalimang
41234ABCDE	ABCEFG0132	1234567894	ClokGoreng	40200000	AB014	Netanyahu	Niagara
51234ABCDE	ABCEFG0123	1234567890	LusiusCS	120500000	AB123	Jl. Pintu Surga Kondangsari	Cirebon
9876543215	ABCEFG0126	1234567894	CirengAyam	40200000	AB127	Jawa Barat	Margonda
9876543216	ABCEFG0125	1234567893	TahuBulat	20200000	AB126	Jawa Barat	Bandung
9876543217	ABCEFG0124	1234567892	ClokGoreng	2200000	AB125	Netanyahu	Niagara
9876543218	ABCEFG0124	1234567891	Abloo	1220000	AB124	Jujutsu	Kalimang
9876543219	ABCEFG0123	1234567890	LusiusCS	120500000	AB123	Jl. Pintu Surga Kondangsari	Cirebon

Output:

#	Time	Action	Message	Duration / Fetch
74	19:36:43	select * from seller LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec
75	19:37:12	select * from transaksi LIMIT 0, 1000	15 row(s) returned	0.016 sec / 0.000 sec

*SELECT **

FROM user;

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator mysqsampledatabase e-shop SQL File 4' x

Limit to 1000 rows

1 select * from user

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

UserID	AlamatEmail	NomorHP
Abloo	Abloo@email.com	08696969691
CikokGoreng	CikokGoreng@email.com	08142042072
CirengAyam	CirengAyam@email.com	08157042062
LusiusCS	LusiusCS@email.com	08123456789
TahuBulat	TahuBulat@email.com	08157042069

Table: user

Columns:

- UserID varchar(25) PK
- AlamatEmail varchar(99)
- NomorHP varchar(13)

Output

#	Time	Action	Message	Duration / Fetch
75	19:37:12	select * from transaksi LIMIT 0, 1000	15 row(s) returned	0.016 sec / 0.000 sec
76	19:37:46	select * from user LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

PENUTUP

Kesimpulan yang dapat kita sampaikan bahwa, kami telah mempelajari dan memahami cara membuat suatu ER-Diagram beserta skema relasinya berdasarkan aturan bisnis yang ada sebagai pengantar dalam pemodelan database. Terima kasih atas waktu yang diberikan untuk membaca laporan kami. Bila ada kesalahan dalam proses pembuatan, mohon diberi kritik dan saran untuk evaluasi kedepannya.