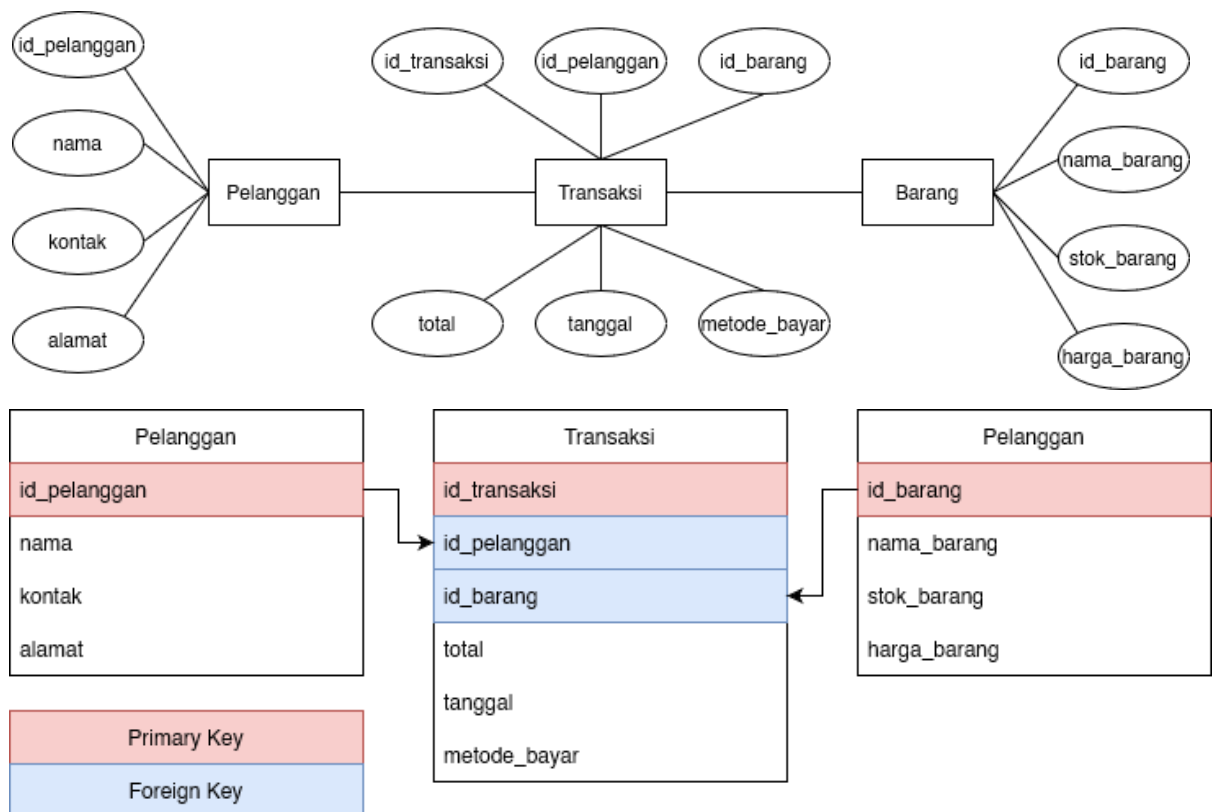


## LEMBAR JAWAB

### UAS Pengolahan Basis Data

Nama = Luksantyanto Dhanie Sadono  
 NIM = 20.01.4543  
 KELAS = 20 D3TI02

1. Dari kasus diatas rancanglah database dengan menggunakan ER-Diagram, secara bertahap, lengkap sampai mapping ER Diagram ke dalam bentuk tabel/ database relasional!



2. Hasil dari Mapping ERD buatlah struktur table yang terbentuk, dengan memperhatikan referential integrity nya yaitu primay key (kunci primer) dan foregin key (kunci tamu)!

Entitas	Atribut
Pelanggan	id_pelanggan, nama, kontak, alamat
Transaksi	Id_transaksi, id_pelanggan, id_barang, total, tanggal, metode_bayar
Barang	Id_barang, nama_barang, stok_barang, harga_barang

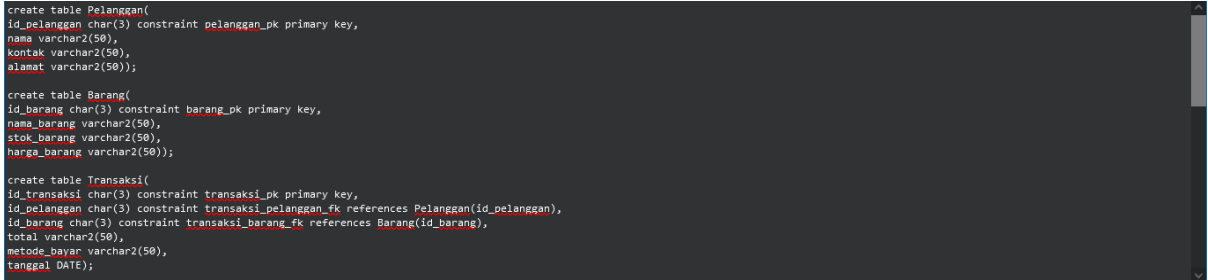
Entitas	Primary Key	Foreign Key
Pelanggan	Id_pelanggan	
Transaksi	Id_transaksi	Id_pelanggan, id_barang
Barang	Id_barang	

3. Tuliskan perintah query untuk membuat konstrain (Foreign key) dari tabel-tabel yang berhubungan.

```
create table Pelanggan(  
id_pelanggan char(3) constraint pelanggan_pk primary key,  
nama varchar2(50),  
kontak varchar2(50),  
alamat varchar2(50));
```

```
create table Barang(  
id_barang char(3) constraint barang_pk primary key,  
nama_barang varchar2(50),  
stok_barang varchar2(50),  
harga_barang varchar2(50));
```

```
create table Transaksi(  
id_transaksi char(3) constraint transaksi_pk primary key,  
id_pelanggan char(3) constraint transaksi_pelanggan_fk references  
Pelanggan(id_pelanggan),  
id_barang char(3) constraint transaksi_barang_fk references Barang(id_barang),  
total varchar2(50),  
metode_bayar varchar2(50),  
tanggal DATE);
```

A screenshot of a SQL code editor with a dark background. The code is written in a light-colored font and includes the same SQL statements as the previous blocks: creating tables Pelanggan, Barang, and Transaksi with their respective columns, primary keys, and foreign key constraints. The code is formatted with line breaks and indentation.

```
create table Pelanggan(  
id_pelanggan char(3) constraint pelanggan_pk primary key,  
nama varchar2(50),  
kontak varchar2(50),  
alamat varchar2(50));  
  
create table Barang(  
id_barang char(3) constraint barang_pk primary key,  
nama_barang varchar2(50),  
stok_barang varchar2(50),  
harga_barang varchar2(50));  
  
create table Transaksi(  
id_transaksi char(3) constraint transaksi_pk primary key,  
id_pelanggan char(3) constraint transaksi_pelanggan_fk references Pelanggan(id_pelanggan),  
id_barang char(3) constraint transaksi_barang_fk references Barang(id_barang),  
total varchar2(50),  
metode_bayar varchar2(50),  
tanggal DATE);
```

4. Tuliskan perintah query untuk menginputkan data pada salah satu tabel serta perintah query untuk mengubah data tersebut dan menghapusnya.

a. Insert

```
insert into Pelanggan  
values('004','Guntur','081501073010','Yogyakarta');
```

```
insert into Barang  
values('B04','USB','20','Rp 10.0000');
```

```
insert into Transaksi  
values('T04','004','B04','Rp 50.000','Bayar Ditempat','23-JAN-2022');
```

```

insert all
into Pelanggan values('001','Charles','08157903909','Yogyakarta')
into Pelanggan values('002','Yoga Djamban','08123456789','Bandung')
into Pelanggan values('003','Charles','081579317901','Surabaya')
select * from dual;

insert all
into Barang values('B01','Keyboard','5','Rp 150.000')
into Barang values('B02','Laptop','12','Rp 2.000.000')
into Barang values('B03','VGA','7','Rp 1.000.000')
select * from dual;

insert all
into Transaksi values('T01','001','B01','Rp 300.000','Transfer','10-JAN-2022')
into Transaksi values('T02','002','B02','Rp 4.000.000','Transfer','13-JAN-2022')
into Transaksi values('T03','003','B03','Rp 3.000.000','Bayar Ditempat','21-JAN-2022')
select * from dual;

```

```

insert into Barang
values('B04','USB','20','Rp 10.000');

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

```

insert into Pelanggan
values('004','Guntur','081501073010','Yogyakarta');

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.01 seconds

```

insert into Transaksi
values('T04','004','B04','Rp 50.000','Bayar Ditempat','23-JAN-2022');

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

## b. Update

update Pelanggan set nama = 'Waluyo' where id\_pelanggan = '003';

```

update Pelanggan set nama = 'Waluyo' where id_pelanggan = '003';

```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

## c. Delete

delete from Transaksi where id\_transaksi = 'T04';

```

delete from Transaksi where id_transaksi = 'T04';

```

Results Explain Describe Saved SQL History

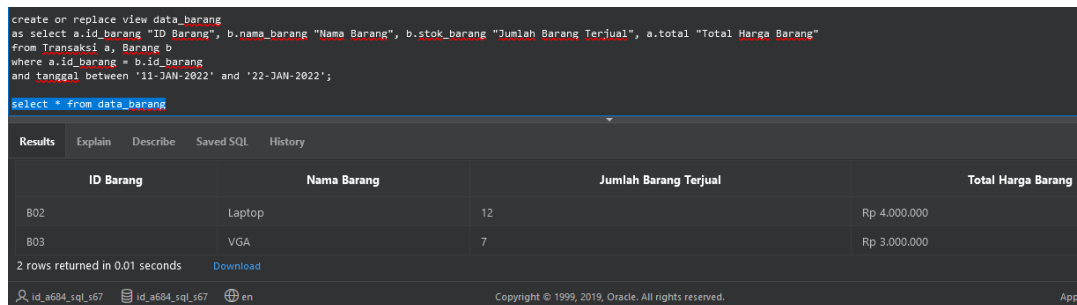
1 row(s) deleted.

0.00 seconds

5. Tuliskan perintah untuk membuat View dari kasus berikut:
- Menampilkan id barang, nama barang, jumlah barang terjual, dan total harga barang yang terjual dengan jangkauan tanggal tertentu!

```
create or replace view data_barang
as select a.id_barang "ID Barang", b.nama_barang "Nama Barang", b.stok_barang
"Jumlah Barang Terjual", a.total "Total Harga Barang"
from Transaksi a, Barang b
where a.id_barang = b.id_barang
and tanggal between '11-JAN-2022' and '22-JAN-2022';
```

```
select * from data_barang
```



The screenshot shows the SQL Developer interface. The top pane contains the SQL code to create a view named 'data\_barang' and a query to select all data from it. The bottom pane shows the results of the query as a table with four columns: ID Barang, Nama Barang, Jumlah Barang Terjual, and Total Harga Barang. Two rows of data are displayed.

```
create or replace view data_barang
as select a.id_barang "ID Barang", b.nama_barang "Nama Barang", b.stok_barang "Jumlah Barang Terjual", a.total "Total Harga Barang"
from Transaksi a, Barang b
where a.id_barang = b.id_barang
and tanggal between '11-JAN-2022' and '22-JAN-2022';
select * from data_barang
```

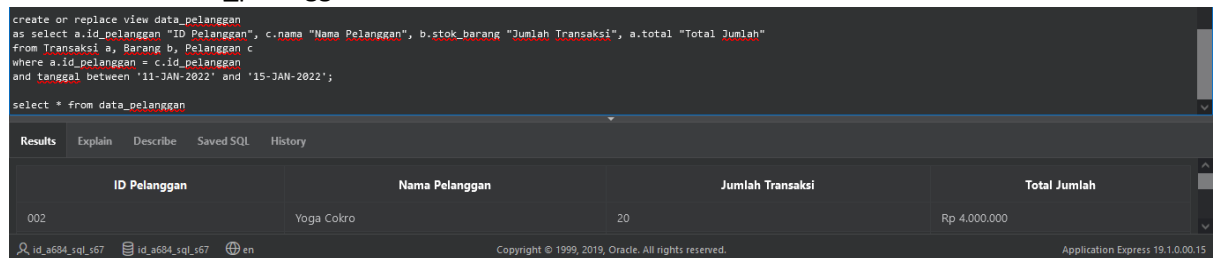
ID Barang	Nama Barang	Jumlah Barang Terjual	Total Harga Barang
B02	Laptop	12	Rp 4.000.000
B03	VGA	7	Rp 3.000.000

2 rows returned in 0.01 seconds

- Menampilkan id pelanggan, nama pelanggan, jumlah transaksi, dan nominal total jumlah yang dibayarkan dengan jangkauan tanggal tertentu!

```
create or replace view data_pelanggan
as select a.id_pelanggan "ID Pelanggan", c.nama "Nama Pelanggan", b.stok_barang
"Jumlah Transaksi", a.total "Total Jumlah"
from Transaksi a, Barang b, Pelanggan c
where a.id_pelanggan = c.id_pelanggan
and tanggal between '11-JAN-2022' and '15-JAN-2022';
```

```
select * from data_pelanggan
```



The screenshot shows the SQL Developer interface. The top pane contains the SQL code to create a view named 'data\_pelanggan' and a query to select all data from it. The bottom pane shows the results of the query as a table with four columns: ID Pelanggan, Nama Pelanggan, Jumlah Transaksi, and Total Jumlah. One row of data is displayed.

```
create or replace view data_pelanggan
as select a.id_pelanggan "ID Pelanggan", c.nama "Nama Pelanggan", b.stok_barang "Jumlah Transaksi", a.total "Total Jumlah"
from Transaksi a, Barang b, Pelanggan c
where a.id_pelanggan = c.id_pelanggan
and tanggal between '11-JAN-2022' and '15-JAN-2022';
select * from data_pelanggan
```

ID Pelanggan	Nama Pelanggan	Jumlah Transaksi	Total Jumlah
002	Yoga Cokro	20	Rp 4.000.000