

# **LAPORAN PRAKTIKUM**

## **PRAKTIKUM DATABASE PERTEMUAN 4**

Disusun untuk Memenuhi Matakuliah Praktikum {Nama Matakuliah}

Dibimbing oleh : Sulaibatul Aslamiyah, M.Kom



Oleh:

**RAFLI RAHMAN.EFENDY**

**1124102162**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**SEKOLAH TINGGI ILMU KOMPUTER PGRI BANYUWANGI**  
**2025**

**LEMBAR PENGESAHAN**  
**LAPORAN PRAKTIKUM**

Matakuliah : Praktikum Database  
Oleh : Rafli Rahman.Efendy  
NIM : 1124102162

**Telah disahkan pada**

Hari : RABU  
Tanggal : 16/03/2025

**Mengetahui/ Menyetujui :**

Dosen Pengampu Mata Kuliah

**Sulaibatul Aslamiah, M.Kom**

NIDN. 0712058304

## **KATA PENGANTAR**

Berisikan kata pengantar yang dituliskan oleh mahasiswa untuk laporan praktikum matakuliah yang diikuti.

## **MODUL PRAKTIKUM 3**

### **1.1 Capaian Praktikum Pertemuan 1**

PEMBUATAN DATABASE KOMPLEKS DAN RELASI TABLE dalam satu cangkupan

### **1.2 Indikator Capaian**

- Membuat Tabel 'db\_Kantor', 'customer', 'Transaksi', dan 'detail\_Transaksi' di buat dalam table mysql tanpa eror.
- Membuat Sruktur table yang sesuai dengan yang di buat/study kasus nya(kolom tipe data, dan constraint).

### **1.3 Landasan Teori**

Jelaskan landasan teori yang digunakan dalam penyelesaian studi kasus. pembuatan tabel - tabel didalam database, dimana antar tabel saling terkait satu sama lainnya

### **1.4 Pelaksanaan Praktikum**

Landasan teori MySQL adalah Structured Query Language (SQL) sebagai bahasa interaktif untuk mengelola data. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL untuk menjalankan fungsinya.

#### 1.4.1 Percobaan Pertama

Terlebih dahulu dibuatkan sebuah database untuk menampung tabel-tabel tersebut. Database yang dibuat diberikan nama db\_toko. Selanjutnya terdapat 4 tabel yang dipergunakan antara lain tabel customer, transaksi, barang, dan detail\_transaksi.

Dengan Membuat database db\_toko: "create database if not exists db\_toko;"

**Di buatkan Table Barang:**

```
mysql> use db_toko;
Database changed
mysql> create table barang(
  -> id_barang VARCHAR(5) NOT NULL,
  -> nama_barang VARCHAR(30) NOT NULL,
  -> satuan VARCHAR(10) NOT NULL,
  -> harga DECIMAL(10,0) NOT NULL DEFAULT 0,
  -> stock INT(3) NOT NULL DEFAULT 0,
  -> PRIMARY KEY (id_barang)
  -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

**Di buatkan Table Costomer:**

```
mysql> use db_toko;
Database changed
mysql> create table customer(
  -> id_customer VARCHAR (5) NOT NULL,
  -> nama_customer VARCHAR (40) NOT NULL,
  -> alamat TEXT NOT NULL,
  -> telepon VARCHAR (12) NOT NULL,
  -> email VARCHAR (50) NOT NULL,
  -> PRIMARY KEY (id_customer)
  -> );
Query OK, 0 rows affected (0.04 sec)
```

**Di buatkan Table Transaksi:**

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO barang (id_barang,nama_barang,satuan,harga,stock) VALUES
  -> ('B001','Buku Tulis','perbiji','2000','90'),
  -> ('B002','Pulpen','buah','3500','70'),
  -> ('B003','Penggaris','perbiji','2000','20'),
  -> ('B004','Pensil','perbiji','2000','40'),
  -> ('B005','Ram','perbiji','90000','40');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

### Di buatkan Detail\_Transaksi:

```
mysql> use db_toko;
Database changed
mysql> create table Transaksi(
  -> id_transaksi INT (6) NOT NULL AUTO_INCREMENT,
  -> id_customer VARCHAR (6) NOT NULL,
  -> tanggal_pesan DATE NOT NULL,
  -> PRIMARY KEY (id_transaksi),
  -> KEY id_customer (id_customer),
  -> CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)
  -> REFERENCES customer (id_customer)
  -> );
Query OK, 0 rows affected, 1 warning (0.14 sec)
```

### Deskripsi :

Untuk menampilkan semua hasil yang telah di coding di Mysql membuat '• create database if not exists db\_toko;' lalu terdapat 4 tabel yang dipergunakan antara lain tabel customer, transaksi, barang, dan detail\_transaksi

### Hasil Uji Coba :

#### Hasil dari Table Barang:

```
mysql> use db_toko;
Database changed
mysql> create table barang(
  -> id_barang VARCHAR(5) NOT NULL,
  -> nama_barang VARCHAR(30) NOT NULL,
  -> satuan VARCHAR(10) NOT NULL,
  -> harga DECIMAL(10,0) NOT NULL DEFAULT 0,
  -> stock INT(3) NOT NULL DEFAULT 0,
  -> PRIMARY KEY (id_barang)
  -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

```
mysql> use db_toko;
Database changed
mysql> use barang;
ERROR 1049 (42000): Unknown database 'barang'
mysql> create table if not exists barang;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> desc barang;
```

Field	Type	Null	Key	Default	Extra
id_barang	varchar(5)	NO	PRI	NULL	
nama_barang	varchar(30)	NO		NULL	
satuan	varchar(10)	NO		NULL	
harga	decimal(10,0)	NO		0	
stock	int	NO		0	

Hasil dari Table Costomer:

```
mysql> use db_toko;
Database changed
mysql> create table customer(
  -> id_customer VARCHAR (5) NOT NULL,
  -> nama_customer VARCHAR (40) NOT NULL,
  -> alamat TEXT NOT NULL,
  -> telepon VARCHAR (12) NOT NULL,
  -> email VARCHAR (50) NOT NULL,
  -> PRIMARY KEY (id_customer)
  -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_customer    | varchar(5)    | NO   | PRI | NULL    |       |
| nama_customer  | varchar(40)   | NO   |     | NULL    |       |
| alamat         | text          | NO   |     | NULL    |       |
| telepon        | varchar(12)   | NO   |     | NULL    |       |
| email          | varchar(50)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Hasil dari Table Transaksi:

```
mysql> use db_toko;
Database changed
mysql> create table Transaksi(
  -> id_transaksi INT (6) NOT NULL AUTO_INCREMENT,
  -> id_customer VARCHAR (6) NOT NULL,
  -> tanggal_pesan DATE NOT NULL,
  -> PRIMARY KEY (id_transaksi),
  -> KEY id_customer (id_customer),
  -> CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)
  -> REFERENCES customer (id_customer)
  -> );
Query OK, 0 rows affected, 1 warning (0.14 sec)
```

```
mysql> desc Transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	int	NO	PRI	NULL	auto_increment
id_customer	varchar(6)	NO	MUL	NULL	
tanggal_pesanan	date	NO		NULL	

Hasil dari Table detail\_transaksi:

```
mysql> use db_toko;
Database changed
mysql> create table detail_transaksi(
  -> id_transaksi INT (6) NOT NULL,
  -> id_barang VARCHAR (6) NOT NULL,
  -> jumlah INT (5) NOT NULL DEFAULT 0,
  -> harga DECIMAL (10,0) NOT NULL DEFAULT 0,
  -> PRIMARY KEY (id_transaksi, id_barang),
  -> KEY FK_detail_transaksi (id_barang),
  -> KEY id_transaksi (id_transaksi),
  -> CONSTRAINT FK_detail_transaksi FOREIGN KEY (id_barang)
  -> REFERENCES barang (id_barang),
  -> CONSTRAINT FK_detail_transaksi2 FOREIGN KEY (id_transaksi)
  -> REFERENCES transaksi (id_transaksi)
  -> );
Query OK, 0 rows affected, 2 warnings (0.10 sec)
```

```
mysql> desc detail_transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	int	NO	PRI	NULL	
id_barang	varchar(6)	NO	PRI	NULL	
jumlah	int	NO		0	
harga	decimal(10,0)	NO		0	

4 rows in set (0.01 sec)

Hasil uji coba untuk menampilkan semua data tersebut tidak di buat satu persatu table melainkan di buat dilalam data “db\_toko”, dan dibuatkan table/desc table lain tabel customer, transaksi, barang, dan detail\_transaksi.



## 1.4.2

### 1. Hasil Uji Coba: Membuat Table Barang beserta Isi didalam nya

```
mysql> use db_toko;
Database changed
mysql> create table barang(
  -> id_barang VARCHAR(5) NOT NULL,
  -> nama_barang VARCHAR(30) NOT NULL,
  -> satuan VARCHAR(10) NOT NULL,
  -> harga DECIMAL(10,0) NOT NULL DEFAULT 0,
  -> stock INT(3) NOT NULL DEFAULT 0,
  -> PRIMARY KEY (id_barang)
  -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

```
mysql> use db_toko;
Database changed
mysql> use barang;
ERROR 1049 (42000): Unknown database 'barang'
mysql> create table if not exists barang;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> desc barang;
```

Field	Type	Null	Key	Default	Extra
id_barang	varchar(5)	NO	PRI	NULL	
nama_barang	varchar(30)	NO		NULL	
satuan	varchar(10)	NO		NULL	
harga	decimal(10,0)	NO		0	
stock	int	NO		0	

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO barang (id_barang,nama_barang,satuan,harga,stock) VALUES
  -> ('B001','Buku Tulis','perbiji','2000','90'),
  -> ('B002','Pulpen','buah','3500','70'),
  -> ('B003','Penggaris','perbiji','2000','20'),
  -> ('B004','Pensil','perbiji','2000','40'),
  -> ('B005','Ram','perbiji','90000','40');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from barang;
```

id_barang	nama_barang	satuan	harga	stock
B001	Buku Tulis	perbiji	2000	90
B002	Pulpen	buah	3500	70
B003	Penggaris	perbiji	2000	20
B004	Pensil	perbiji	2000	40
B005	Ram	perbiji	90000	40

5 rows in set (0.00 sec)

### Deskripsi :

Untuk menampilkan semua hasil yang telah di coding di Mysql membuat 'create table barang' lalu table yang telah di buat ketik 'use db\_toko;' agar bisa di gunakan untuk mengisi table database yang sudah di buat.

### Hasil Uji Coba :

Hasil uji coba untuk menampilkan data siswa dengan menggunakan perintah 'desc barang;' akan menampilkan seluruh table yang telah di buat, untuk menampilkan apa yang sudah di isi didalam table 'select \* from barang;'

### 2. Hasil Uji Coba: Membuat Table Customer beserta Isi didalam nya

```
mysql> use db_toko;
Database changed
mysql> create table customer(
  -> id_customer VARCHAR (5) NOT NULL,
  -> nama_customer VARCHAR (40) NOT NULL,
  -> alamat TEXT NOT NULL,
  -> telepon VARCHAR (12) NOT NULL,
  -> email VARCHAR (50) NOT NULL,
  -> PRIMARY KEY (id_customer)
  -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> desc customer;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_customer    | varchar(5)    | NO   | PRI | NULL     |       |
| nama_customer  | varchar(40)   | NO   |     | NULL     |       |
| alamat         | text          | NO   |     | NULL     |       |
| telepon        | varchar(12)   | NO   |     | NULL     |       |
| email          | varchar(50)   | NO   |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> INSERT INTO customer VALUES
  -> ('P001', 'Ahmad Sutopo', 'Jln Kepiting 45 Banyuwangi', '254654', 'a.sutopo@email.com'),
  -> ('P002', 'Indah Lestari', 'Jln. Setaman 32 Banyuwangi', '443426', 'indah54@email.com');
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO customer (id_customer,nama_customer,alamat,telepon,email) VALUES
  -> ('P003', 'Budi Suprpto', 'Jln. Suropati 56 Banyuwangi', '663452', 'budi.s@email.com'),
  -> ('P004', 'Hendrik Maulana', 'Jln. Merdeka 13 Banyuwangi', '351854', 'hendrikjabrik@email.com'),
  -> ('P005', 'Rahmat Arif', 'Jln. Jaya Giri 14 Banyuwangi', '35627', 'arif99@email.com');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from customer;
```

id_customer	nama_customer	alamat	telepon	email
P001	Ahmad Sutopo	Jln Kepiting 45 Banyuwangi	254654	a.sutopo@email.com
P002	Indah Lestari	Jln. Setaman 32 Banyuwangi	443426	indah54@email.com
P003	Budi Suprpto	Jln. Suropati 56 Banyuwangi	663452	budi.s@email.com
P004	Hendrik Maulana	Jln. Merdeka 13 Banyuwangi	351854	hendrikjabrik@email.com
P005	Rahmat Arif	Jln. Jaya Giri 14 Banyuwangi	35627	arif99@email.com

```
5 rows in set (0.00 sec)
```

### Deskripsi :

Untuk menampilkan semua hasil yang telah di coding di Mysql membuat 'create table customer' lalu table yang telah di buat ketik 'use db\_toko;' agar bisa di gunakan untuk mengisi table database yang sudah di buat.

### Hasil Uji Coba :

Hasil uji coba untuk menampilkan data siswa dengan menggunakan perintah 'desc costomer;' akan menampilkan seluruh table yang telah di buat, untuk menampilkan apa yang sudah di isi didalam table 'select \* from costomer;'

### 3. Hasil Uji Coba: Membuat Table Transaksi beserta Isi didalam nya

```
mysql> use db_toko;
Database changed
mysql> create table Transaksi(
  -> id_transaksi INT (6) NOT NULL AUTO_INCREMENT,
  -> id_customer VARCHAR (6) NOT NULL,
  -> tanggal_pesan DATE NOT NULL,
  -> PRIMARY KEY (id_transaksi),
  -> KEY id_customer (id_customer),
  -> CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)
  -> REFERENCES customer (id_customer)
  -> );
Query OK, 0 rows affected, 1 warning (0.14 sec)
```

```
mysql> desc Transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	int	NO	PRI	NULL	auto_increment
id_customer	varchar(6)	NO	MUL	NULL	
tanggal_pesan	date	NO		NULL	

```
mysql> use db_toko;
Database changed
mysql> create table Transaksi(
  -> id_transaksi INT (6) NOT NULL AUTO_INCREMENT,
  -> id_customer VARCHAR (6) NOT NULL,
  -> tanggal_pesan DATE NOT NULL,
  -> PRIMARY KEY (id_transaksi),
  -> KEY id_customer (id_customer),
  -> CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)
  -> REFERENCES customer (id_customer)
  -> );
Query OK, 0 rows affected, 1 warning (0.14 sec)
```

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO Transaksi (id_transaksi,id_customer,tanggal_pesanan) VALUES
-> ('1','P001','2017-10-10'),
-> ('2','P002','2017-10-12'),
-> ('4','P004','2017-10-16'),
-> ('3','P002','2017-10-12'),
-> ('5','P001','2017-10-17');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from Transaksi;
+-----+-----+-----+
| id_transaksi | id_customer | tanggal_pesanan |
+-----+-----+-----+
| 1 | P001 | 2017-10-10 |
| 2 | P002 | 2017-10-12 |
| 3 | P002 | 2017-10-12 |
| 4 | P004 | 2017-10-16 |
| 5 | P001 | 2017-10-17 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### Deskripsi :

Untuk menampilkan semua hasil yang telah di coding di Mysql membuat 'create table Transaksi' lalu table yang telah di buat ketik 'use db\_toko;' agar bisa di gunakan untuk mengisi table database yang sudah di buat.

#### Hasil Uji Coba :

Hasil uji coba untuk menampilkan data siswa dengan menggunakan perintah 'desc costomer;' akan menampilkan seluruh table yang telah di buat, untuk menampilkan apa yang sudah di isi didalam table 'select \* from Transaksi;'

#### 4. Hasil Uji Coba: Membuat Table detail\_transaksi beserta Isi didalam nya

```
mysql> use db_toko;
Database changed
mysql> create table detail_transaksi(
-> id_transaksi INT (6) NOT NULL,
-> id_barang VARCHAR (6) NOT NULL,
-> jumlah INT (5) NOT NULL DEFAULT 0,
-> harga DECIMAL (10,0) NOT NULL DEFAULT 0,
-> PRIMARY KEY (id_transaksi, id_barang),
-> KEY FK_detail_transaksi (id_barang),
-> KEY id_transaksi (id_transaksi),
-> CONSTRAINT FK_detail_transaksi FOREIGN KEY (id_barang)
-> REFERENCES barang (id_barang),
-> CONSTRAINT FK_detail_transaksi2 FOREIGN KEY (id_transaksi)
-> REFERENCES transaksi (id_transaksi)
-> );
Query OK, 0 rows affected, 2 warnings (0.10 sec)
```

```
mysql> desc detail_transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	int	NO	PRI	NULL	
id_barang	varchar(6)	NO	PRI	NULL	
jumlah	int	NO		0	
harga	decimal(10,0)	NO		0	

```
4 rows in set (0.01 sec)
```

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO detail_transaksi (id_transaksi,id_barang,jumlah,harga) VALUES
-> ('1','B002','1','3000'),
-> ('1','B003','1','3000'),
-> ('2','B004','3','2000'),
-> ('3','B001','1','2700'),
-> ('4','B002','2','3000'),
-> ('5','B004','1','2700');
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO detail_transaksi (id_transaksi,id_barang,jumlah,harga) VALUES
-> ('1','B002','1','3000'),
-> ('1','B003','1','3000'),
-> ('2','B004','3','2000'),
-> ('3','B001','1','2700'),
-> ('4','B002','2','3000'),
-> ('5','B004','1','2700');
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from detail_transaksi;
```

id_transaksi	id_barang	jumlah	harga
1	B002	1	3000
1	B003	1	3000
2	B004	3	2000
3	B001	1	2700
4	B002	2	3000
5	B004	1	2700

```
6 rows in set (0.00 sec)
```

#### Deskripsi :

Untuk menampilkan semua hasil yang telah di coding di Mysql membuat 'create table detail\_transaksi' lalu table yang telah di buat ketik 'use db\_toko;' agar bisa di gunakan untuk mengisi table database yang sudah di buat.

#### Hasil Uji Coba :

Hasil uji coba untuk menampilkan data siswa dengan menggunakan perintah 'desc costomer;' akan menampilkan seluruh table yang telah di buat, untuk menampilkan apa yang sudah di isi didalam table 'select \* from detail\_transaksi;'

### 1.4.3 Tugas Praktikum Database Pada Modul 3

Terlebih dahulu dibuatkan sebuah database untuk menampung tabel-tabel tersebut. Database yang dibuat diberikan nama db\_toko. Selanjutnya terdapat 4 tabel yang dipergunakan antara lain tabel customer, transaksi, barang, dan detail\_transaksi. Query dalam pembuatan tabel-tabel tersebut dijabarkan sebagai berikut:

#### Membuat database db toko

- create database if not exists db\_toko;

#### # Membuat Create table customer:

```
mysql> use db_toko;
Database changed
mysql> create table customer(
    -> id_customer VARCHAR (5) NOT NULL,
    -> nama_customer VARCHAR (40) NOT NULL,
    -> alamat TEXT NOT NULL,
    -> telepon VARCHAR (12) NOT NULL,
    -> email VARCHAR (50) NOT NULL,
    -> PRIMARY KEY (id_customer)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

#### Membuat table customer

- create table customer (  
    id\_customer VARCHAR (5) NOT NULL,  
  
    nama\_customer VARCHAR (40) NOT NULL,  
  
    alamat TEXT NOT NULL,  
  
    telepon VARCHAR (20) NOT NULL,  
  
    email VARCHAR (50) NOT NULL,  
  
    PRIMARY KEY (id\_customer)  
  
)

### # Membuat Create table Barang:

```
mysql> use db_toko;
Database changed
mysql> create table barang(
  -> id_barang VARCHAR(5) NOT NULL,
  -> nama_barang VARCHAR(30) NOT NULL,
  -> satuan VARCHAR(10) NOT NULL,
  -> harga DECIMAL(10,0) NOT NULL DEFAULT 0,
  -> stock INT(3) NOT NULL DEFAULT 0,
  -> PRIMARY KEY (id_barang)
  -> );
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

#### Membuat table barang

- create table barang (  
    id\_barang VARCHAR (5) NOT NULL,  
  
    nama\_barang VARCHAR (30) NOT NULL,  
  
    satuan VARCHAR (10) NOT NULL,  
  
    harga DECIMAL (10,0) NOT NULL DEFAULT '0',  
  
    stock INT (3) NOT NULL DEFAULT '0',  
  
    PRIMARY KEY (id\_barang)  
  
    )

### # Membuat Create table Transaksi:

```
mysql> use db_toko;
Database changed
mysql> create table Transaksi(
  -> id_transaksi INT (6) NOT NULL AUTO_INCREMENT,
  -> id_customer VARCHAR (6) NOT NULL,
  -> tanggal_pesan DATE NOT NULL,
  -> PRIMARY KEY (id_transaksi),
  -> KEY id_customer (id_customer),
  -> CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)
  -> REFERENCES customer (id_customer)
  -> );
Query OK, 0 rows affected, 1 warning (0.14 sec)
```

#### Membuat table transaksi

- create table transaksi (  
    id\_transaksi INT (5) NOT NULL AUTO\_INCREMENT,  
  
    id\_customer VARCHAR (5) NOT NULL,  
  
    tgl\_pesan DATE NOT NULL,  
  
    PRIMARY KEY (id\_transaksi),

```

KEY id_customer (id_customer),

CONSTRAINT transaksi_ibfk_1 FOREIGN KEY (id_customer)

REFERENCES customer (id_customer)

)

```

#### # Membuat Create table detail\_transaksi:

```

mysql> use db_toko;
Database changed
mysql> create table detail_transaksi(
-> id_transaksi INT (6) NOT NULL,
-> id_barang VARCHAR (6) NOT NULL,
-> jumlah INT (5) NOT NULL DEFAULT 0,
-> harga DECIMAL (10,0) NOT NULL DEFAULT 0,
-> PRIMARY KEY (id_transaksi, id_barang),
-> KEY FK_detail_transaksi (id_barang),
-> KEY id_transaksi (id_transaksi),
-> CONSTRAINT FK_detail_transaksi FOREIGN KEY (id_barang)
-> REFERENCES barang (id_barang),
-> CONSTRAINT FK_detail_transaksi2 FOREIGN KEY (id_transaksi)
-> REFERENCES transaksi (id_transaksi)
-> );
Query OK, 0 rows affected, 2 warnings (0.10 sec)

```

#### Membuat table detail transaksi

- create table detail\_transaksi (
 id\_transaksi INT (5) NOT NULL,

 id\_barang VARCHAR (5) NOT NULL,

 jumlah INT (5) NOT NULL DEFAULT '0',

 harga DECIMAL (10,0) NOT NULL DEFAULT '0',

 PRIMARY KEY (id\_transaksi, id\_barang),

 KEY FK\_detail\_transaksi (id\_barang),

 KEY id\_transaksi (id\_transaksi),

 CONSTRAINT FK\_detail\_transaksi FOREIGN KEY (id\_barang)

 REFERENCES barang (id\_barang),

 CONSTRAINT FK\_detail\_transaksi2 FOREIGN KEY (id\_transaksi)

 REFERENCES transaksi (id\_transaksi)

 )



### Deskripsi :

Perhatikan pada tabel transaksi dan tabel detail\_transaksi, dimana pada tabel transaksi terdapat satu field yang mengandung Foreign Key yaitu id\_customer yang berhubungan dengan tabel customer. Dan pada tabel detail\_transaksi terdapat 2 Foreign Key yaitu id\_barang yang terhubung dengan tabel barang dan id\_transaksi yang terhubung dengan tabel transaksi.

### Hasil Uji Coba :

Meyatukan semua table yang di hubungkan dalam satu table

#### # DESC Customer

```
mysql> desc customer;
```

Field	Type	Null	Key	Default	Extra
id_customer	varchar(5)	NO	PRI	NULL	
nama_customer	varchar(40)	NO		NULL	
alamat	text	NO		NULL	
telepon	varchar(12)	NO		NULL	
email	varchar(50)	NO		NULL	

5 rows in set (0.01 sec)

#### # DESC Barang

```
mysql> use db_toko;
Database changed
mysql> use barang;
ERROR 1049 (42000): Unknown database 'barang'
mysql> create table if not exists barang;
ERROR 4028 (HY000): A table must have at least one visible column.
mysql> desc barang;
```

Field	Type	Null	Key	Default	Extra
id_barang	varchar(5)	NO	PRI	NULL	
nama_barang	varchar(30)	NO		NULL	
satuan	varchar(10)	NO		NULL	
harga	decimal(10,0)	NO		0	
stock	int	NO		0	

#### # DESC Transaksi

```
mysql> desc Transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	int	NO	PRI	NULL	auto_increment
id_customer	varchar(6)	NO	MUL	NULL	
tanggal_pesanan	date	NO		NULL	

## # DESC detail\_transaksi

```
mysql> desc detail_transaksi;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_transaksi | int | NO | PRI | NULL | |
| id_barang | varchar(6) | NO | PRI | NULL | |
| jumlah | int | NO | | 0 | |
| harga | decimal(10,0) | NO | | 0 | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

### Hasil Uji Coba 1 :

kolom atau field pada suatu tabel yang berfungsi sebagai kunci tamu dari tabel lain. Foreign Key sangat berguna bila bekerja dengan banyak tabel yang saling berelasi satu sama lain. Isikan data pada setiap tabel tersebut. Query yang dipergunakan adalah sebagai berikut:

## # Membuat atau mengisi table Insert Data Customer

```
mysql> INSERT INTO customer VALUES
-> ('P001','Ahmad Sutopo','Jln Kepiting 45 Banyuwangi','254654','a.sutopo@email.com'),
-> ('P002','Indah Lestari','Jln. Setaman 32 Banyuwangi','443426','indah54@email.com');
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO customer (id_customer,nama_customer,alamat,telepon,email) VALUES
-> ('P003','Budi Suprpto','Jln. Suropati 56 Banyuwangi','663452','budi.s@email.com'),
-> ('P004','Hendrik Maulana','Jln. Merdeka 13 Banyuwangi','351854','hendrikjabrik@email.com'),
-> ('P005','Rahmat Arif','Jln. Jaya Giri 14 Banyuwangi','35627','arif99@email.com');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- INSERT INTO customer VALUES  
( 'P001', 'Ahmad Sutopo', 'Jln Kepiting 45 Banyuwangi', '254654',  
'a.sutopo@email.com'),  
( 'P002', 'Indah Lestari', 'Jln. Setaman 32 Banyuwangi', '443426',  
'indah54@email.com');

## # Membuat atau mengisi table Insert Data Barang

```
mysql> use db_toko;
Database changed
mysql> INSERT INTO barang (id_barang,nama_barang,satuan,harga,stock) VALUES
-> ('B001','Buku Tulis','perbiji','2000','90'),
-> ('B002','Pulpen','buah','3500','70'),
-> ('B003','Penggaris','perbiji','2000','20'),
-> ('B004','Pensil','perbiji','2000','40'),
-> ('B005','Ram','perbiji','90000','40');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

- INSERT INTO barang (id\_barang, nama\_barang, satuan, harga, stock) VALUES  
( 'B001', 'Buku Tulis', 'perbiji', '2000', '90'),  
( 'B002', 'Pulpen', 'buah', '3500', '70'),  
( 'B003', 'Penggaris', 'perbiji', '2000', '20'),

```
('B004', 'Pensil', 'perbiji', '2000', '40'),  
('B005', 'Ram', 'perbiji', '90000', '40');
```

### # Membuat atau mengisi table Insert Data Transaksi

```
mysql> use db_toko;  
Database changed  
mysql> INSERT INTO Transaksi (id_transaksi, id_customer, tanggal_pesan) VALUES  
-> ('1', 'P001', '2017-10-10'),  
-> ('2', 'P002', '2017-10-12'),  
-> ('4', 'P004', '2017-10-16'),  
-> ('3', 'P002', '2017-10-12'),  
-> ('5', 'P001', '2017-10-17');  
Query OK, 5 rows affected (0.02 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

- INSERT INTO Transaksi (id\_transaksi, id\_customer, tanggal\_pesan) VALUES  
( '1', 'P001', '2017-10-10'),  
( '2', 'P002', '2017-10-12'),  
( '4', 'P004', '2017-10-16'),  
( '3', 'P002', '2017-10-12'),  
( '5', 'P001', '2017-10-17');

### # Membuat atau mengisi table Insert Data detail\_transaksi

```
mysql> use db_toko;  
Database changed  
mysql> INSERT INTO detail_transaksi (id_transaksi, id_barang, jumlah, harga) VALUES  
-> ('1', 'B002', '1', '3000'),  
-> ('1', 'B003', '1', '3000'),  
-> ('2', 'B004', '3', '2000'),  
-> ('3', 'B001', '1', '2700'),  
-> ('4', 'B002', '2', '3000'),  
-> ('5', 'B004', '1', '2700');  
Query OK, 6 rows affected (0.01 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

- INSERT INTO detail\_transaksi (id\_transaksi, id\_barang, jumlah, harga) VALUES  
( '1', 'B002', '1', '3000'),  
( '1', 'B003', '1', '3000'),  
( '2', 'B004', '3', '2000'),  
( '3', 'B001', '1', '2700'),  
( '4', 'B002', '2', '3000'),  
( '5', 'B004', '1', '2700');

### Deskripsi :

Perhatikan pada tabel transaksi dan tabel detail\_transaksi, dimana pada tabel transaksi terdapat satu field yang mengandung Foreign Key yaitu id\_customer yang berhubungan dengan tabel customer. Dan pada tabel detail\_transaksi terdapat 2 Foreign Key yaitu

id\_barang yang terhubung dengan tabel barang dan id\_transaksi yang terhubung dengan tabel transaksi.

### Hasil Uji Coba :

Meyatukan semua table yang di hubungkan dalam satu table

# select \* from costomer

```
mysql> select * from customer;
```

id_customer	nama_customer	alamat	telepon	email
P001	Ahmad Sutopo	Jln Kepiting 45 Banyuwangi	254654	a.sutopo@email.com
P002	Indah Lestari	Jln. Setaman 32 Banyuwangi	443426	indah54@email.com
P003	Budi Suprpto	Jln. Suropati 56 Banyuwangi	663452	budi.s@email.com
P004	Hendrik Maulana	Jln. Merdeka 13 Banyuwangi	351854	hendrikjabrik@email.com
P005	Rahmat Arif	Jln. Jaya Giri 14 Banyuwangi	35627	arif99@email.com

5 rows in set (0.00 sec)

# select \* from Barang

```
mysql> select * from barang;
```

id_barang	nama_barang	satuan	harga	stock
B001	Buku Tulis	perbiji	2000	90
B002	Pulpen	buah	3500	70
B003	Penggaris	perbiji	2000	20
B004	Pensil	perbiji	2000	40
B005	Ram	perbiji	90000	40

5 rows in set (0.00 sec)

# select \* from Transaksi

```
mysql> select * from Transaksi;
```

id_transaksi	id_customer	tanggal_pesan
1	P001	2017-10-10
2	P002	2017-10-12
3	P002	2017-10-12
4	P004	2017-10-16
5	P001	2017-10-17

5 rows in set (0.00 sec)

# select \* from detail\_transaksi

```
mysql> select * from detail_transaksi;
```

id_transaksi	id_barang	jumlah	harga
1	B002	1	3000
1	B003	1	3000
2	B004	3	2000
3	B001	1	2700
4	B002	2	3000
5	B004	1	2700

6 rows in set (0.00 sec)

Di dalam suatu RDBMS, dalam satu database dapat terdiri dari beberapa tabel. Masing-masing tabel tersebut berhubungan satu sama lain atau dengan kata lain memiliki relasi.

Relasi antartabel dapat berupa relasi 1-1, 1-M, atau M-N. Sebagai contoh terlihat pada gambar pemodelan data konseptual (class diagram) di atas. Tabel customer berhubungan dengan transaksi, transaksi dengan barang, dan sebagainya. Pada praktisnya, terkadang dibutuhkan tampilan data dari beberapa tabel sekaligus. Misal, diinginkan untuk menampilkan nama customer dengan transaksi yang pernah dilakukannya. Dari contoh tersebut, dapat dilakukan penggabungan minimal dua tabel, yaitu customer dan transaksi. Untuk menggabungkan 2 (dua) atau lebih tabel, dapat menggunakan bentuk perintah JOIN.

#### 1.4.4 Tugas Praktikum Database Pada Modul 3

##### Inner Join

Dengan inner join, tabel akan digabungkan dua arah, sehingga tidak ada data yang NULL di satu sisi. Cara yang pertama dengan menggunakan where. Berikut syntax perintahnya:

##### Where

```
mysql> SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi, transaksi.tanggal_pesan FROM customer, Transaksi
WHERE customer.id_customer = Transaksi.id_customer;
```

id_customer	nama_customer	id_transaksi	tanggal_pesan
P001	Ahmad Sutopo	1	2017-10-10
P002	Indah Lestari	2	2017-10-12
P002	Indah Lestari	3	2017-10-12
P004	Hendrik Maulana	4	2017-10-16
P001	Ahmad Sutopo	5	2017-10-17

5 rows in set (0.00 sec)

- Menggabung dua tabel dengan kondisi Where:

SELECT tabel1.\*, tabel2.\* FROM tabel1, tabel2 WHERE tabel1.PK=tabel2.FK;

Atau

SELECT customer.id\_customer, customer.nama\_customer, Transaksi.id\_transaksi,  
transaksi.tanggal\_pesan FROM customer, Transaksi  
WHERE customer.id\_customer Transaksi.id\_customer;

##### Inner Join

Jika dengan menggunakan Inner Join, maka query perintahnya adalah sebagai berikut:

```
mysql> SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi, transaksi.tanggal_pesan FROM customer INNER JOIN
Transaksi ON customer.id_customer = Transaksi.id_customer;
```

id_customer	nama_customer	id_transaksi	tanggal_pesan
P001	Ahmad Sutopo	1	2017-10-10
P002	Indah Lestari	2	2017-10-12
P002	Indah Lestari	3	2017-10-12
P004	Hendrik Maulana	4	2017-10-16
P001	Ahmad Sutopo	5	2017-10-17

5 rows in set (0.00 sec)

- Menggabung dua tabel dengan Inner Join:

SELECT tabel1.\*, tabel2.\* FROM tabel1 INNER JOIN tabel2 ON tabel1.PK=tabel2.FK;

Atau

```
SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi,
transaksi.tanggal_pesan FROM customer INNER JOIN
Transaksi ON customer.id_customer = Transaksi.id_customer;
```

## Outer Join

Dengan outer join, tabel akan digabungkan satu arah, sehingga memungkinkan terdapat data yang NULL (kosong) di satu sisi. Sebagai contoh semisal akan menampilkan daftar pelanggan yang pernah melakukan transaksi. Outer join terbagi menjadi dua yaitu right join dan left join. Perintah query:

### Left join

```
mysql> SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi, Transaksi.tanggal_pesan
-> FROM customer
-> LEFT JOIN Transaksi ON customer.id_customer = Transaksi.id_customer;
```

id_customer	nama_customer	id_transaksi	tanggal_pesan
P001	Ahmad Sutopo	5	2017-10-17
P001	Ahmad Sutopo	1	2017-10-10
P002	Indah Lestari	3	2017-10-12
P002	Indah Lestari	2	2017-10-12
P003	Budi Suprpto	NULL	NULL
P004	Hendrik Maulana	4	2017-10-16
P005	Rahmat Arif	NULL	NULL

7 rows in set (0.01 sec)

- Menggabung tabel dengan left join:

```
SELECT tabel1.*, tabel2.* FROM tabel1 LEFT JOIN tabel2 ON tabel1.PK=tabel2.FK;
```

Atau

```
SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi,
Transaksi, tanggal_pesan
FROM customer
LEFT JOIN Transaksi ON customer.id_customer = Transaksi.id_customer;
```

### Right Join

```
mysql> SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi, transaksi.tanggal_pesan FROM customer RIGHT JOIN
Transaksi ON customer.id_customer = transaksi.id_customer;
```

id_customer	nama_customer	id_transaksi	tanggal_pesan
P001	Ahmad Sutopo	1	2017-10-10
P002	Indah Lestari	2	2017-10-12
P002	Indah Lestari	3	2017-10-12
P004	Hendrik Maulana	4	2017-10-16
P001	Ahmad Sutopo	5	2017-10-17

5 rows in set (0.01 sec)

- Menggabung tabel dengan right Join

```
SELECT tabel1.*, tabel2.* FROM tabel1 RIGHT JOIN tabel2 ON tabel1.PK=tabel2.FK;
```

Atau

```
SELECT customer.id_customer, customer.nama_customer, Transaksi.id_transaksi,  
transaksi.tanggal_pesan FROM customer RIGHT JOIN  
Transaksi ON customer.id_customer = transaksi.id_customer;
```

### 3.3 Soal Latihan Soal Modul 3 Pertemuan 3

Tampilkan nama barang serta jumlah barang yang telah terjual dari tabel **transaksi** dan **barang**. (menggunakan format **Where** atau Join). Hasil tabel seperti berikut:

```
mysql> SELECT barang.nama_barang, detail_transaksi.jumlah  
-> FROM barang, detail_transaksi  
-> WHERE barang.id_barang = detail_transaksi.id_barang  
-> AND detail_transaksi.jumlah > 0;  
+-----+-----+  
| nama_barang | jumlah |  
+-----+-----+  
| Pulpen      | 1      |  
| Penggaris   | 1      |  
| Pensil      | 3      |  
| Buku Tulis  | 1      |  
| Pulpen      | 2      |  
| Pensil      | 1      |  
+-----+-----+  
6 rows in set (0.01 sec)
```

```
SELECT barang.nama_barang, detail_transaksi.jumlah  
FROM barang, detail_transaksi  
WHERE barang.id_barang = detail_transaksi.id_barang  
AND detail_transaksi.jumlah > 0;
```

Gabungkan 3 tabel (**transaksi**, **detail\_transaksi**, dan **barang**) agar dapat menampilkan barang yang dipesan, nama, barang, serta harga untuk pemesanan dengan id transaksi = 1. (menggunakan format **Where** atau **Join**). Hasil tabel seperti berikut:

```
mysql> SELECT transaksi.id_transaksi,  
-> detail_transaksi.id_barang,  
-> barang.nama_barang,  
-> barang.harga,  
-> detail_transaksi.jumlah  
-> FROM transaksi, detail_transaksi, barang  
-> WHERE transaksi.id_transaksi = detail_transaksi.id_transaksi  
-> AND detail_transaksi.id_barang = barang.id_barang  
-> AND transaksi.id_transaksi = 1;  
+-----+-----+-----+-----+  
| id_transaksi | id_barang | nama_barang | harga | jumlah |  
+-----+-----+-----+-----+  
| 1            | B002     | Pulpen      | 3500  | 1      |  
| 1            | B003     | Penggaris   | 2000  | 1      |  
+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

```

SELECT transaksi.id_transaksi,
detail transaksi.id_barang,
barang.nama_barang,
barang.harga,
-> detail_transaksi.jumlah
-> FROM transaksi, detail_transaksi, barang
-> WHERE transaksi.id_transaksi = detail_transaksi.id_transaksi
-> AND detail_transaksi.id_barang = barang.id_barang
AND transaksi.id_transaksi = 1;

```

Gabungkan 3 tabel (**customer**, **transaksi**, dan **detail\_transaksi**) agar dapat menampilkan data customer serta jumlah **transaksi** yang telah dilakukan oleh customer tersebut.

(menggunakan format **Where** atau **Join**). Hasil tabel seperti berikut:

```

mysql> SELECT customer.nama_customer, COUNT(transaksi.id_transaksi) AS jumlah
-> FROM customer, transaksi, detail_transaksi
-> WHERE customer.id_customer = transaksi.id_customer
-> AND transaksi.id_transaksi = detail_transaksi.id_transaksi
-> GROUP BY customer.id_customer;
+-----+-----+
| nama_customer | jumlah |
+-----+-----+
| Ahmad Sutopo  | 3      |
| Indah Lestari | 2      |
| Hendrik Maulana | 1     |
+-----+-----+
3 rows in set (0.01 sec)

```

```

SELECT customer.nama_customer, COUNT(transaksi.id_transaksi) AS jumlah
-> FROM customer, transaksi, detail_transaksi
-> WHERE customer.id_customer = transaksi.id_customer
AND transaksi.id_transaksi = detail_transaksi.id_transaksi
GROUP BY customer.id_customer;

```

Apa perbedaan penggabungan tabel menggunakan kondisi where dan join? Dan apa perbedaan antara left join dan right join? Jelaskan berdasarkan pengertian anda sendiri beserta query pembuatan tabel diatas. Diusahakan tidak plagiarism / kesamaan artikel.

---JAWABAN---

- **Perbedaan Penggabungan Tabel dengan WHERE dan JOIN**

Secara fungsional, JOIN dan penggabungan menggunakan WHERE bisa memberikan hasil yang sama, khususnya untuk INNER JOIN, tapi ada beberapa perbedaan penting:

JOIN	WHERE
------	-------



Lebih terstruktur dan eksplisit dalam menggabungkan tabel.	Kurang eksplisit, karena kondisi penggabungan ditulis di bagian WHERE bersama kondisi filter.
Mendukung berbagai jenis join seperti LEFT JOIN, RIGHT JOIN, dll.	Biasanya hanya digunakan untuk INNER JOIN.
Lebih mudah dibaca untuk query kompleks.	Bisa jadi membingungkan ketika melibatkan banyak kondisi.

- **Perbedaan LEFT JOIN dan RIGHT JOIN**

LEFT JOIN	RIGHT JOIN
Mengambil semua data dari tabel kiri, dan mencocokkan dengan tabel kanan. Jika tidak cocok, data dari kanan jadi NULL.	Mengambil semua data dari tabel kanan, dan mencocokkan dengan tabel kiri. Jika tidak cocok, data dari kiri jadi NULL.
SELECT * FROM A LEFT JOIN B → semua dari A tetap tampil.	SELECT * FROM A RIGHT JOIN B → semua dari B tetap tampil.

**Link Github Saya:**

<https://github.com/PwRafli/SP2.1-Praktikum-Database>

## **BAB 1**

### **PENUTUP**

#### **Kesimpulan**

teori MySQL adalah Structured Query Language (SQL) sebagai bahasa interaktif untuk mengelola data. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL untuk menjalankan fungsinya..

#### **Saran**

Saran saya tidak ada perubahan pembelajaran sama seperti si Semester 1 dan masih masuk di praktikum kali kecuali pada pengisian laporan praktikum database ke word itu yang perlu di tanyakan dan cara pengisian laporan untuk kedepannya.

## DAFTAR PUSTAKA

Tuliskan rujukan yang anda gunakan baik website maupun buku seperti contoh dibawah.

1. Tim Asisten Dosen. 2014. Modul 1 Pengenalan Sistem Operasi, Ide Visual C++, Dan Algoritma Pemrograman. Malang: Universitas Negeri Malang.
2. Program Konversi Suhu (online)  
<http://bondanoky.blogspot.com/2012/10/program-konversi-suhu-c.html>. Di akses 8 September.

## **LAMPIRAN**

Berisikan syntax atau gambar yang dibutuhkan dalam tiap pertemuan praktikum.